Position and Force Control of Cooperating Robots Using Inverse Dynamics

Du, Zhenyu

Award date:
2015

Awarding institution:
University of Bath

Position and Force Control of Cooperating Robots Using Inverse Dynamics

Zhenyu Du

A thesis submitted for the degree of Doctor of Philosophy

University of Bath

Department of Mechanical Engineering

October 2014

Abstract

Multiple robot manipulators cooperating in a common manipulation task can accomplish complex tasks that a single manipulator would be unable to complete. To achieve physical cooperation with multiple manipulators working on a common object, interaction forces need to be controlled throughout the motion. The aim of this research is to develop an inverse dynamics model-based cooperative force and position control scheme for multiple robot manipulators. An extended definition of motion is proposed to include force demands based on a constrained Lagrangian dynamics and Lagrangian multipliers formulation. This allows the direct calculation of the inverse dynamics with both motion and force demands. A feedforward controller based on the proposed method is built to realise the cooperative control of two robots sharing a common load, with both motion and force demands. Furthermore, this thesis develops a method to design an optimal excitation trajectory for robot dynamic parameter estimation utilising the Schroeder Phased Harmonic Sequence. This method yields more precise and accurate inverse dynamics models, which result in better control. The proposed controller is then tested in an experimental set-up consisting of two robot manipulators and a common load. Results show that in general the proposed controller performs noticeably better position and force tracking, especially for higher speed motions, when compared to traditional hybrid position/force controllers.

Acknowledgement

I would like to take this opportunity to express my gratitude to my supervisors Prof. M.Necip Sahinkaya, Dr. Pejman Iravani, and Prof. Andrew Plummer for their inspiring guidance and invaluable support throughout my research.

I would like to thank my parents for their emotional and financial support. Without their support, my education would have been impossible.

I would also like to thank my wife for her understanding and company during my study in Bath.

I wish to extend my gratitude to all my friends and colleagues at the Centre for Power Transmission and Motion Control (CPTMC) and the Robotics Lab at the University of Bath for their help and support. I would also like to thank sincerely Judy and Robin for their friendly support.

This thesis is dedicated to my wife Qian, and our children Jiayu and Qihang.

Contents

Nomenclature

$\mathbf{0}$	Vector or matrix of zeros
α	Time scaling parameter
ζ	Damping ratio
$\boldsymbol{\Theta}$	Vector of parameters
θ	Parameter
θ	Absolute joint angle
ε	Normalized time variable
$\boldsymbol{\tau}$	Vector of torques
φ	Phase
ω	Frequency
ω_n	Natural frequency
λ	Lagrangian multiplier
$\boldsymbol{\lambda}$	Vector of Lagrangian multipliers
$\mathbf{A}$	Square matrix containing **M** and **J**
A	Amplitude
$\mathbf{B}$	Matrix of motion defining generalised coordinates
b	Analogue to digital output
C	Correlation coefficient
$\mathbf{C}$	Vector of centrifugal and Coriolis forces/torques
c	Friction coefficient
$\mathbf{D}$	Vector containing centrifugal, Coriolis and gravity forces and torques
E	Kinetic energy function
F	Force
$\mathbf{F}$	Vector of forces
f	Constraint equation

$\mathbf{f}$	Vector of constraint equations
$\mathbf{G}$	Vector of gravity forces/torques
g	Gravitational acceleration
H	Positive integer
h	Estimated simulation step size
I	Moment of inertia
i	Index
$\mathbf{J}$	Constraint Jacobian matrix
j	Index
K	Number of degrees of freedom of a motion
k	Stiffness
$\boldsymbol{L}$	Lagrangian function
L	Length
M	Number of degrees of freedom of a system
$\mathbf{M}$	Mass and inertia matrix
m	Mass
N	Number of generalised coordinates
N_{H}	Number of frequency components
n	Integer
P	Link centre of gravity distance to previous axis
P_{i}, P_{j}	Power ratios
Q	Generalised input
$\mathbf{Q}$	Vector of generalised inputs
q	Generalised coordinate
$\mathbf{q}$	Vector of generalised coordinates
T	Torque
T_s	Desired settling time

T	Vector of torques
t	Time
u	Generalised control input
u	Vector of generalised control inputs
V	Potential energy function
v	Velocity
v	Vector of superfluous coordinates
W	Work
x	Cartesian coordinate
Y	Regression matrix
y	Cartesian coordinate
y	Desired motion acceleration functions
z	Cartesian coordinate
DOF	Degrees of freedom
FMFF	Force and motion feedforward
HIL	Hardware in the loop
MFF	Motion-only feedforward
SPHS	Schroeder Phased Harmonic Sequence

1 Introduction

1.1 Overview

Since the concept of cooperative robotic systems was first seriously explored in the 1980s, the systems themselves have rapidly evolved from the first dual axes robot arms to the complex and highly specified robots used in industrial settings today [1-6]. Cooperative robotics is a multi-disciplinary field that spans the full range of computer science, electrical engineering, artificial intelligence, biology, communication, and robotics [7]. Two or more robots cooperating in a single manipulation task can accomplish different kinds of complex task that a single manipulator would be unable to complete alone. Even when a single robot is able to achieve a given task, it is possible that employing a group of robots can improve the performance of the overall system [8]. Furthermore, single multi-link actuators, such as a robotic hand, require the cooperative actuation of all its parts.

With the ever-increasing development of the modern world, the demand for everyday products drives industries to invest more and more in automation. In 2013, an “all-time high” of almost 180,000 industrial robots were sold worldwide, which is 12 percent more than in 2012 [9]. Continuing development of the manipulators and control systems that allows cooperative systems to be effective at controlling forces and able to accomplish increasingly detailed and specific tasks is necessary for implementation of these systems in industry.

Controlling the forces of manipulators in cooperatively controlled robots is a challenge because the forces in three-dimensional spaces may act independently on different agents in the system. As soon as multiple manipulators grip one object, the robotic system forms a closed chain mechanism that is extremely nonlinear and coupled, which has an effect on the controller design. With only position-controlled robots, it is problematic to accomplish a synchronised trajectory of two or more robots with no error between them. Small errors in the trajectories will result in huge internal force which can damage the load and the robots themselves, therefore it is necessary to regulate the internal force [10]. To achieve this, one can use force sensors

and hence force feedback control. However, when moving fast, a simple PI controlled force feedback loop cannot handle the nonlinearities of the manipulator system. Consequently, more advanced force control methods are demanded. Addressing force control in cooperating manipulator systems is vital to understanding cooperative control in current industrial robotics and in prototypes that may become part of robotics technology in the future.

1.2 Hypothesis and Contributions

The hypothesis explored in this thesis is the following:

“The usage of Lagrangian methods to derive feedforward controllers that combine position and force will improve the control of fast moving cooperative robotic systems.”

The cooperative robotic systems referred to are two or more robots sharing a common physical constraint; for example, two robots carry a shared load.

The primary contribution of the thesis is to introduce a formulation of the Lagrangian dynamics to be applied to force- and position-constrained tasks that occur during multi-robot physical interaction. Chapter 3 describes the manipulation of the constrained equations such that force and position can be used as hybrid inputs to the model. This is in contrast to the conventional approach in which only acceleration is used as the demand.

The second contribution is to introduce a novel approach to optimally identify model parameters experimentally. Chapter 4 describes the use of Schroeder Phased Harmonic Sequence (SPHS) in obtaining optimal excitation trajectories. SPHS has fewer parameters to optimise compared to the conventional Fourier series or polynomial curves.

1.3 Objectives of the Research

The objectives to prove the hypothesis are:

- To develop an inverse dynamics model-based cooperative manipulator control scheme that regulates internal force while achieving desired motion. Based on constrained Lagrangian dynamics and Lagrangian

multipliers, an extended definition of motion inputs is proposed to include force demands. This allows direct calculation of the inverse dynamics with both motion and force inputs. Furthermore, given the generality of the Lagrangian formulation, different physical phenomena can be modelled within the unified approach, enabling the modelling of complex systems.

- To implement the proposed controller experimentally. A test rig consisting of two robot manipulators is used for the studying of the controller performance.
- To develop a method of optimal excitation trajectory design for the estimation of robot dynamic parameters. Existing methods require many parameters during the optimisation process. The proposed method should use fewer parameters.

1.4 Layout of the Thesis

This thesis is organised as follows: Chapter 2 presents a review of past literature that covers the cooperation of multiple robots, robot force control and its application in cooperative manipulation, and trajectory design for dynamic parameter estimation of robot manipulators. Chapter 3 describes the derivation of the extended inverse dynamics formulation and the feedforward control scheme. Simulations are presented on cooperative controlled manipulators. Chapter 4 presents the hardware and software used in the research and a brief experiment set-up. Then the new method to design optimal excitation trajectories is presented. Chapter 5 and 6 present a detailed set-up for each experimental case and the results are reported and analysed. Chapter 7 presents the conclusions and recommendations for future developments.

2 Background to Multi-Robot Systems and Force Control

By definition, industrial robots are generally reprogrammable, multipurpose machines with some ability to sense, plan, and act to complete defined functions, and they generally have a contained mechanism for observing and addressing disturbance. These robots can often carry out motion in six axes and utilise a broad range of end-effectors. They comprise one of the largest and most diversified classes of machines that are able to interact autonomously in an industrial environment. Additional challenges arise when programming for complex or specific processes among multiple robotic systems.

The coordinated control of multiple robot manipulators has been actively studied by many researchers. Since robots have become an integral part of human life, they are presented with increasingly difficult tasks. Many of these tasks can be achieved by multiple robots better than a single robot. By cooperating, robots can complete tasks more quickly, improve system robustness, and achieve tasks impossible for a single robot. A robotic system consisting of several manipulators has more capability than a single manipulator for tasks such as handling heavy material and assembly.

Controlling force in a varied group of robots provides a unique challenge for engineers and researchers. To achieve cooperation when multiple manipulators hold a common object, internal and external forces need to be measured and controlled. With only position-controlled robots, it is problematic to accomplish the synchronised trajectory of two robots with no error between them [11]. A small error in the trajectories will result in huge internal forces which can damage the load; therefore, it is necessary to measure and control internal forces.

An accurate dynamic model of the manipulator is required for the purposes of model-based control and robot manipulator simulations. Known kinematic and inertial parameters of the manipulator can be used to construct a dynamic model. The kinematic parameters, such as joint types, link lengths, and offsets between joint axes can be obtained from the manufacturer, or from calibrations,

without difficulty. However, most of the time inertial parameters such as link mass, location of the centre of gravity (COG), and moment of inertia, are not available from the manufacturers. Therefore, many attempts have been made to identify these parameters.

This chapter is organised as follows: Section 2.1 presents a review of multi-robot cooperation and classification; Section 2.2 presents a review of position and force control; Section 2.3 presents a review of robot dynamic parameter estimation.

2.1 Multi-Robot Cooperation

Multi-robot system technology has progressed rapidly from simulation, to laboratory prototyping, to realization of real-world applications [12]. Cooperatively controlled industrial robots, among others, have become capable of greater operating work volumes and better repeatability accuracy in the past decades, as well as dropping to a price range practical in many industrial settings. A typical industrial robot that cost $100,000 ten years ago, now costs $30,000, making it much more attractive to medium sized manufacturing and industrial companies [4]. While multi-robot systems offer many advantages and increased potential when compared to single robots, there are still many challenges in their design, realization, and control that must be overcome in order to yield cost-effective and efficient multi-robot systems.

The quality and control of robotic manipulators has improved. However, simple arm joints are still widely used, with joints that are actuated by electric motors or hydraulic actuators, though newer arm geometries and joints allow for easier maintenance and more detailed, reprogrammable motion. Improvements in robot control, planning, and human interfacing are necessary to allow cooperative robot systems to be a practical alternative as manufacturing continues to shift from the high volume assembly line and conveyor belts of the mid-twentieth century, to the contemporary High-Mix Low-Volume (HMLV) manufacturing seen in current industrial environments. Advances in coordinated robotic systems control coupled with more capable manipulators have made these systems more practical for industry [13].

When two or more manipulators handle one object, they form a closed chain mechanism, which is referred to as a parallel mechanism. A parallel mechanism is where “the stationary link and the output link are connected by more than one link” [14]. Manipulators that consist of parallel mechanisms are termed parallel manipulators [15]. Furthermore, when there are more actuators than the degrees of freedom (DOF) of the system, it is then called redundantly actuated parallel manipulators [16].

The cooperative manipulators have close resemblance to parallel manipulators. They share similar kinematic and dynamic properties. However, they serve different purposes hence their research focuses are different. Cooperative manipulators consist of individual manipulators which carry out tasks together, possibly with different loads from time to time. Therefore, the connections between manipulators and loads are temporary and consequently the dynamics of the mechanisms change. The research of cooperative manipulators focuses on coordination control, motion planning, robustness, force control and mobile applications [17-23]. On the other hand, parallel manipulators have built-in closed chain mechanisms and their goals are often to provide moving platforms or interact with other objects. The research of redundant parallel manipulators focuses mainly on position accuracy improvement, force capacity improvement and force distribution [24-26]. Internal forces considered by redundantly actuated parallel manipulators are preloads which can eliminate backlash in the system [27].

Researchers generally agree that multi-robot systems have several advantages over single robot systems [28-30]. The most common motivations for developing multi-robot system solutions are that: (1) the task complexity is too high for a single robot to accomplish; (2) the task is inherently distributed; (3) building several resource-bounded robots is much easier than having a single powerful robot; (4) multiple robots can solve problems faster using parallelism; and (5) the introduction of multiple robots increases robustness through redundancy. Examples of manipulation tasks that cooperating manipulators may achieve include automated manufacturing, assembling, and handling large, heavy or flexible objects [31-36]. The types of robot considered in this thesis are those cooperating robot arms, such as in load handling.

To understand multi-robot systems as a whole, a classification of multi-robot systems is necessary. Many researchers have proposed various taxonomies for these systems. Cao et al. proposed five research axes on cooperative behaviour: group architectures, resource conflicts, origins of cooperation, learning, and geometric problems [37]. Dudek et al. provided a classification and metrics scheme for these systems by classifying robotic collectives by seven different characteristics: collective size, range of communication, communication topology, communication bandwidth, collective reconfigurability, processing ability of each agent, and collective composition [38]. Gerkey and Matarić provided a taxonomy of task allocation problems based on three classes: single/multi task robots, single/multi robot tasks, and instantaneous/time-extended assignments [39]. Winfield proposed a taxonomy focusing on robot foraging with four major axes (environment, robot(s), performance and strategy) and 19 minor axes [40].

However, the taxonomy that is most relevant to this thesis is the one provided by Farinelli et al. [41]. The authors proposed a taxonomy of coordinated multi-robot systems. A hierarchical structure is illustrated in Figure 2.1. Although focused on complex mobile platforms, the taxonomy is applicable to other platforms.

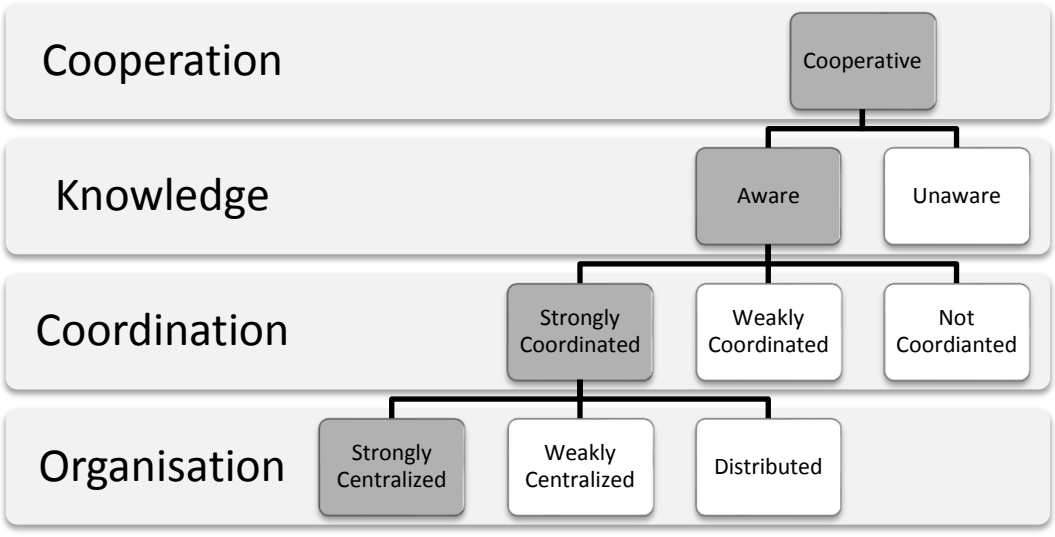

Figure 2.1 Multi-robot systems taxonomy [41], with shaded blocks indicating the classification of this thesis

The proposed controller in this thesis is designed for multiple robot cooperative load manipulation. The controller needs the dynamic information of all robots

and load. All feedforward and feedback control is executed by a single controller. Therefore, the work focused in this thesis can be classified in Cooperative - Aware - Strongly coordinated - Strongly centralized, as show in Figure 2.1 with the shaded blocks.

2.2 Position and Force Control

De Schutter et al. stated in their bird's eye view of force control that major force control methods were distinguished by their constraint models: hybrid/parallel position/force control deals with geometric constraints; impedance control deals with dynamic constraints [42]. Whereas Zen and Hemami provided a more structured and detailed overview of robot force control [43]. A classification is provided according to their taxonomy in Table 2.1. However, there is a slight alteration to their taxonomy. According to Zen and Hemami, in the fundamental force control, the stiffness, impedance, and admittance control is listed separately. Here they are grouped together under indirect force control for more clarity.

Table 2.1 A classification of robot force control based on taxonomy in [43]

Fundamental force control	Indirect force control	Stiffness control	[44]
		Impedance control	[45-50]
		Admittance control	
	Hybrid control	Hybrid position/force control	[29, 30, 51-57]
		Hybrid impedance control	[58]
	Explicit force control		[59]
	Implicit force control		
Advanced force control	Adaptive force control	Adaptive compliant motion control	[60]
		Adaptive impedance or admittance control	[61, 62]
		Adaptive position/force control	[63-65]
		Adaptive explicit force control	
	Robust force control	Robust compliant motion control	
		Robust impedance or admittance control	[66-68]
		Robust position/force control	[65, 69]
		Robust explicit force control	
	Learning algorithm		[70-73]
	Neural network techniques		[74, 75]
	Fuzzy control		[75-77]

The two most basic approaches are hybrid position/force control and impedance control. Hybrid position/force control regulates position and force in orthogonally separate frames [10, 30, 51, 78]. This approach can be naturally applied to many tasks, such as deburring and polishing. On the other hand, impedance

control does not control position and force directly. Alternatively, it regulates the mechanical impedance of the manipulator, which is the ratio of force to velocity [45, 79, 80]. The main difference between the two approaches is whether the force feedback loop is explicitly enclosed [12].

Both the hybrid position/force control and impedance control started on controlling external forces, *i.e.* when a robot comes into contact with its environment. However, in cooperative manipulation, internal force also needs to be controlled. Tension and compression forces are examples of internal forces. Both hybrid position/force control and impedance control can be applied to cooperative manipulation [46, 47, 54, 77, 81-83], but this process is not straightforward because of the increasing system complexity. Schneider and Cannon used object impedance control to explicitly control internal forces in cooperative manipulation [46]. The differences between three controllers were also compared: coordinated joint PD, coordinated endpoint impedance and object impedance. Bonitz and Hsia applied impedance control to internal force. Force feedback was transformed to trajectory correction through a compliance function [47]. Leksono et al. proposed an impedance control on cooperative motion with variable compliance gain [82]. Perdereau and Drouin considered each cooperative manipulator as an autonomous, independently controlled system and used stiffness like force control [54]. This simplifies the implementation of the coordinated controller. Although the above three groups of authors achieved good force and position tracking, the exact following of the desired force profile is impossible using impedance control strategies [84]. In [81] only the interaction forces at manipulator end points are controlled, not the internal force. The internal force is a subset of the interaction forces that do not contribute to any of the acceleration of the object. Tinos et al. studied cooperative manipulators with passive joints with decomposed motion and squeeze force controls [83]. Inertia matrices are not required by the controller; however, a new Jacobian matrix has to be built. Gueaieb used robust adaptive control for cooperative manipulator tasks in order to cope with dynamic model uncertainties by assuming a closed chain model [77].

In recent years, compliant devices were used on traditional industrial robots. Lange et al. employed a compliant force sensor to reduce force errors when approaching unexpected objects [85]. Dai and Kerr designed and tested a force

measuring device based on the six DOF mini Stewart platform [86]. Lopes and Almeida developed a similar active auxiliary device [87]. It can be mounted on an industrial manipulator and used to execute the impedance and force control laws. Whereas Kim et al. achieved simultaneous force and position control through a mechanical method [56]. The authors built a new actuator unit composed of two actuators and a planetary gear train connected in series, where one high-torque low-speed motor controls the position while the other low-torque high-speed motor regulates the stiffness.

Khatib proposed a unified framework for motion and force control of robot manipulators [78, 88, 89]. Given tasks were specified by the generalised task specification matrix and an active force control term was unified into the operational space command vector. Yoshikawa presented a similar approach with explicitly described end-effector constraint hypersurfaces [52, 90]. In [91], Yoshikawa and Zheng proposed a dynamic hybrid controller for position and force control of multiple manipulators handling a common object. Although the dynamics of the robots and the object were taken into account, a complicated nonlinear state feedback law was employed as the control technique. Frictions and gravity were not included in the robot dynamics.

The controller proposed in this thesis utilises Lagrangian methods to simplify the calculation of the control inputs. An extended motion definition is the key to combine both motion and force demand in inverse dynamics calculation.

2.3 Parameter Estimation

Parameter estimation has been an active area of research for decades [92]. Many methods have been used to model and estimate parameters of robotic systems [93-96]. There are several major factors, such as model formation, estimation algorithm, and excitation trajectory, which can affect the outcome of the process. Section 4.3 presents a new approach to design excitation trajectories for estimating inertial parameters of a robot manipulator.

There are three major methods in terms of robot inertial parameter estimation: physical experiments; CAD models; identification [97]. The first method involves disassembling the robot and obtaining the parameters of individual links. For example, mass can be weighed; COG can be measured by counterbalancing;

pendulous motions can be used to determine the diagonal elements of the inertia tensor. However, this method should ideally be employed by the manufacturer before assembling the robot because disassembly and assembly are labour intensive and technically challenging tasks. The second method needs a CAD model of the robot and sometimes that is simply not available (as in the case of this thesis). Even if the CAD model is available, the method is likely to produce errors because of the complicated nature of robot hardware and often ignored small component parts. Thirdly, the identification method is based on the behaviour analysis of the robot input and output signals, during or after a predefined motion. This method is widely used and tested and therefore it is chosen to obtain the robot parameters later in this thesis.

There are two major approaches in the identification of robot parameters: sequential testing and overall trajectory optimisation [92], both of which have their pros and cons. The sequential test method uses several trajectories which excite different sets of parameters at a time. However, errors accumulate due to the reuse of the parameters from preceding estimations. The overall trajectory optimisation methods often involve solving nonlinear optimisation problems with a large number of parameters. There are various formulations of overall trajectory optimisation process, but these are cumbersome and do not guarantee a global optimum solution.

Armstrong et al. [98] described an approach in which the DOF are points of a series of joint accelerations. This general approach results in a large number of DOF, such that optimisation becomes intractable. Gautier and Khalil [99] optimised a linear combination of the condition number and the equilibrium of the set of equations that generate the parameters. The DOF are a finite set of joint angles and velocities separated in time. The actual trajectory is continuous and smooth, and is calculated by interpolating a fifth-order polynomial between the optimised points, assuming zero initial and final acceleration. However, only a very small part of the final trajectory, the finite set of joint angles and velocities, is optimised. As a result, the total smooth trajectory cannot be guaranteed to satisfy all motion constraints or to be optimal with respect to the condition number or the covariance matrix criterion.

Swevers et al. [100, 101] presented a robot excitation that is periodic. The excitation trajectory for each joint is a finite Fourier series. This approach guarantees improving the quality of the measured signals by time-domain averaging; estimating the noise characteristics without performing additional measurements; and calculating more accurate and less noise sensitive joint velocities and accelerations, which are required to calculate the identification matrices. This approach, however, still results in a relatively large number of trajectory parameters to optimise, depending on the number of harmonics chosen in the Fourier series.

In Section 4.3, a new approach is proposed toward the design of robot excitation trajectories, which is based on the Schroeder Phased Harmonic Sequence (SPHS). This approach will overcome the drawbacks mentioned above. The SPHS signal has fewer parameters; furthermore, using this method guarantees optimal trajectories.

2.4 Concluding Remarks

Following the above review of multi-robot position and force control, this thesis will focus on the development of an inverse dynamics controller for multi-robot motion and force control on single object with tight cooperation. The controller should be simple to implement with all dynamics of the robots and object taken into account. Detailed derivation is described in Chapter 3.

In estimating manipulator dynamic parameters, the previous excitation trajectory design methods mainly use polynomial curves or finite Fourier series as base trajectories. Then the parameters of these trajectories are optimised according to criteria related to identification matrices. However, these trajectories often comprise a large number of parameters, which prolongs the optimisation and may cause the optimisation to end up in local minimums. These problems are easily overcome by the use of SPHS signals proposed in this thesis because SPHS signals consist of fewer parameters. More details can be found in Section 4.3.

3 Extended Inverse Dynamics Controller

In this chapter, a novel method is introduced and derived to calculate inverse dynamics. This method deals with both motion and force inputs at the same time. The advantage of this approach is a simplified calculation of inverse dynamics that integrates force and position demands within the same framework. The conventional method needs to separate the system properly according to the demanded forces. Once separated, the inverse dynamics need to be calculated on each subsystem. This is particularly difficult when the system is complicated, so more human input is needed. The new method involves the construction of a feedforward controller from inverse dynamics.

The chapter is organised as follows: Section 3.1 introduces the new method for calculating inverse dynamics with both motion and force inputs; Section 3.2 presents the control strategy using the proposed inverse dynamics calculation; Section 3.3 presents the simulations of a simplified system using the new controller. Section 3.4 concludes this chapter.

3.1 Description of an Extended Inverse Dynamics Controller

Conventional inverse dynamics formulations do not allow motions with higher DOF than that of the system. This eliminates the use of inverse dynamics for redundant systems, where there are more actuators than the DOF of the system. Even if a manipulator is fully actuated, it may become redundant when cooperating with two or more manipulators to perform a joint task, or there may be external constraints imposed on the motion. In this situation, the proposed extended inverse dynamics is able to calculate control inputs with higher DOF motions.

In this section, the inverse dynamics of robot manipulators is derived, starting with developing differential equations to describe the relationship between forces/torques and motion. Both forward dynamics and inverse dynamics are derived in the following subsections.

3.1.1 Lagrangian Dynamics

Dynamic analysis is important for the design of robots and control algorithms [102]. Newton's three laws and the concept of virtual work may be regarded as the foundation of classical mechanics. However, the basic laws of dynamics can be formulated in several ways other than that given by Newton, such as d'Alembert's principle, Lagrange's equations, Hamilton's equations and Hamilton's principle. All are basically equivalent, but vary in terms of ease of developing the equations of motion. Although Newton's equations are convenient for simple cases, Lagrange's equations offer significant advantages when dealing with multi-body and/or multi-physics mechatronics systems and widely used in robotics. The first part of this section will cover Lagrange's equations, also referred as Euler-Lagrange equations, to describe the system dynamics.

There are two general types of dynamical problems. Almost every problem in classical dynamics is a special case of one of the following general types:

- Forward dynamics: Allows to find the "motion" of the system (*i.e.* the position, velocity and acceleration of each mass as a function of time) from the given forces and torques acting on the system, constraints, and known position and velocity of each mass at a given instant of time (e.g. initial conditions).
- Inverse dynamics: Allows to calculate a possible set of forces and torques as a function of time to produce a specified motion.

The Lagrangian dynamics offer many advantages over conventional Newton's method of writing equations of motion. These include:

- For a large class of mechanical systems, the Lagrange equations provide a unique and sufficiently simple method of constructing equations of motion that is independent of the form (complexity) of the actual system.
- Only work and energy are used, which are scalar quantities and have the same unit for any branches of physics whether mechanical, electrical or chemical.
- The chief advantage of the Lagrange equations is that the number of equations is equal to the number of DOF of the system and is independent of the number of points and bodies in the system.

- Internal forces or any forces not contributing any work are not needed in the derivation. This is a great advantage over Newton's reaction force balancing method for any slightly complicated system. Nevertheless, the forces can still be calculated when required.
- Lagrange's equations take the same form for any coordinate system, so that the method of solution proceeds in the same way for any problem.
- It is invariant under coordinate transformations.
- Only positions and velocities but not accelerations (unlike Newton's method) are needed in the derivation.

3.1.1.1 Lagrangian Formulation

Using Lagrange's equation of motion [102], the dynamic model of an n-DOF rigid body manipulator is derived. It can be stated as:

$$\frac{d}{dt}\left(\frac{\partial \boldsymbol{L}}{\partial \dot{q}_i}\right) - \frac{\partial \boldsymbol{L}}{\partial q_i} = Q_i, \qquad i = 1 \cdots N \tag{3.1}$$

where:

q_i, i = 1 … N are the generalised coordinates

Q_i, i = 1 … N are the generalised inputs (forces/torques)

N is the number of generalised coordinates (normally equal to the number of DOF of the system M)

Defining $\mathbf{q}$ and $\dot{\mathbf{q}}$ as the vectors of generalised coordinates and derivatives, respectively

$\boldsymbol{L}(\mathbf{q}, \dot{\mathbf{q}}, t)$ is the Lagrangian function $\boldsymbol{L} = E - V$

$E(\mathbf{q}, \dot{\mathbf{q}}, t)$ is the kinetic energy function

$V(\mathbf{q}, t)$ is the potential energy function

When Lagrange's equations of motion are developed for any manipulator, the resulting dynamic equations will be in the following general form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{Q} \tag{3.2}$$

where $\mathbf{M}(\mathbf{q})$ is the $N \times N$ mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is an $N \times 1$ vector of centrifugal and Coriolis forces/torques, $\mathbf{G}(\mathbf{q})$ is an $N \times 1$ vector of gravity forces/torques, and $\mathbf{Q}$ is the vector of generalised inputs. If $\mathbf{M}$ is not constant, there may be a position where the corresponding $\mathbf{q}$ values cause the determinant of $\mathbf{M}$ to become zero. This is known as singularity, and there is no solution for the equations of the motion at the singular position.

3.1.1.2 Generalised Coordinates

A great variety of coordinates can be employed as generalised coordinates. Consequently, the choice of generalised coordinates specifies generalised inputs. For example, in a system with a two-link manipulator with a point-mass load, as shown in Figure 3.1, if the angles θ_1 and θ_2 are selected as the generalised coordinates, then the torques τ_1 and τ_2 acting on θ_1 and θ_2 become the generalised inputs. However, if, say x_3 and y_3 are used as generalised coordinates, then the forces acting in the direction of x_3 and y_3, F_x and F_y respectively, become the generalised inputs.

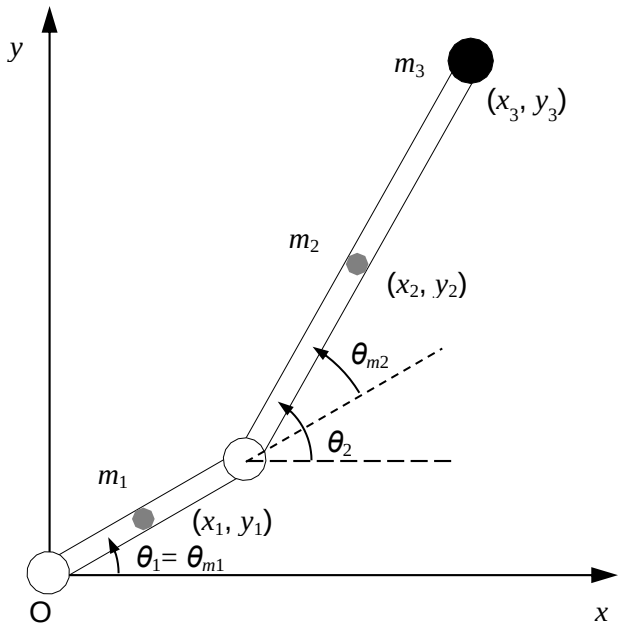

Figure 3.1 A system of a two-link manipulator with a point-mass load

3.1.1.3 Generalised Inputs

There may be external forces acting on the system which are not along the direction of generalised coordinates. One method of expressing these forces along the generalised coordinates is to utilise the principle of virtual work.

The work done by forces $F_{vi}, i = 1 \cdots H$, acting in infinitesimal displacements $v_i, i = 1 \cdots H$, is:

$$\delta W = \sum_{i=1}^{H} F_{vi} \delta v_i \tag{3.3}$$

If the system has N DOF represented by N generalised coordinates $q_i, i = 1 \cdots N$, the coordinates (v_i's) can be eliminated by using the constraint or transformation equations as follows:

$$v_i = v_i(q_1, q_2, \cdots, q_N), \qquad i = 1 \cdots H \tag{3.4}$$

This gives

$$\delta v_i = \sum_{j=1}^{N} \frac{\delta v_i}{\delta q_j} \delta q_j, \qquad i = 1 \cdots H \tag{3.5}$$

Substituting (3.5) into (3.3) gives the total virtual work in the following general form:

$$\delta W = \underbrace{[\cdots]}_{Q_1} \delta q_1 + \underbrace{[\cdots]}_{Q_2} \delta q_2 + \cdots + \underbrace{[\cdots]}_{Q_N} \delta q_N \tag{3.6}$$

where the brackets represents the generalised inputs for the Lagrange's equation of motion.

Take the two-link manipulator in Figure 3.1 as an example, the absolute angles are used as generalised coordinates when developing the equations of motion. However, the system inputs are the torques generated by the two motors driving each joint. Therefore, the generalised inputs in terms of the motor torques T_{m1}

and T_{m2} need to be calculated. Motor torque inputs T_{m1} and T_{m2} act on the relative angles $\theta_{m1} = \theta_1$ and $\theta_{m2} = \theta_2 - \theta_1$, respectively. The total virtual work

$$\delta W = T_{m1}\delta\theta_{m1} + T_{m2}\delta\theta_{m2} \tag{3.7}$$

Replacing motor coordinates with generalised coordinates (*i.e.* absolute angles) gives

$$\delta W = T_{m1}\delta\theta_{m1} + T_{m2}(\delta\theta_2 - \delta\theta_1) \tag{3.8}$$

or

$$\delta W = [T_{m1} - T_{m2}]\delta\theta_1 + [T_{m2}]\delta\theta_2 \tag{3.9}$$

Hence the generalised inputs:

$$Q_1 = T_{m1} - T_{m2}, and\ Q_2 = T_{m2} \tag{3.10}$$

or

$$\mathbf{Q} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \mathbf{T}_m \tag{3.11}$$

3.1.1.4 Constrained Lagrangian Equations

When developing equations of motion, it may be convenient to use more generalised coordinates than the number of DOF, *i.e.* $N > M$. Since only M independent variable can exist for an M - DOF system, N - M constraint equations are needed:

$$f_j(q_1, q_2, \cdots, q_N) = 0, \qquad j = 1 \cdots (N - M) \tag{3.12}$$

or in vector form

$$\mathbf{f}(\mathbf{q}) = [f_1 \cdots f_{N-M}]^T = \mathbf{0} \tag{3.13}$$

These are called holonomic constraints, *i.e.* it is possible to eliminate N - M superfluous coordinates by using (3.12). Of course, superfluous coordinates can be eliminated from the Lagrangian function before developing the Lagrange's equations of motion. However, this may increase the complexity of the algebraic manipulations. The advantages of using superfluous coordinates can be appreciated if the motion were specified in the workspace, as there would be no need to convert the motion into joint coordinates.

The following alternative version of the Lagrange's equations of motion allows the use of more generalised coordinates than the number of DOF:

$$\frac{d}{dt}\left(\frac{\partial \boldsymbol{L}}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} + \sum_{j=1}^{N-M} \lambda_j \frac{\partial f_j}{\partial q_i} = Q_i, i = 1 \cdots N \tag{3.14}$$

where f_j and λ_j are the jth constraint equation and Lagrangian multiplier, respectively, for i = 1… N and j = 1… (N - M). This would produce N equations of motions in the following general form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{J}^T(\mathbf{q})\boldsymbol{\lambda} = \mathbf{Q} \tag{3.15}$$

where $\boldsymbol{\lambda}$ is the vector of Lagrangian multipliers. The matrix $\mathbf{J}$ is the (N - M) × N constraint Jacobian matrix containing the partial derivatives of constraint equations with respect to generalised coordinates.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \cdots & \frac{\partial f_1}{\partial q_N} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_{N-M}}{\partial q_1} & \frac{\partial f_{N-M}}{\partial q_2} & \cdots & \frac{\partial f_{N-M}}{\partial q_N} \end{bmatrix} \tag{3.16}$$

and

$$\boldsymbol{\lambda} = [\lambda_1 \; \lambda_2 \; \cdots \; \lambda_{N-M}] \tag{3.17}$$

Equation (3.15) provides N second order differential equations. Together with N - M constraint equations in (3.13), these provide enough differential-algebraic

equations to solve for $2N$ - M variables (N generalised coordinates and N - M Lagrangian multipliers).

To combine algebraic and differential equations (3.13) and (3.15), the algebraic constraint equations (3.13) can be differentiated twice.

Differentiating once:

$$\mathbf{J}\dot{\mathbf{q}} = \mathbf{0} \tag{3.18}$$

Differentiating twice:

$$\mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} = \mathbf{0} \tag{3.19}$$

Adding this to (3.15) gives the following equations of motion

$$\begin{bmatrix} \mathbf{M}(\mathbf{q}) & \mathbf{J}^T(\mathbf{q}) \\ \mathbf{J}(\mathbf{q}) & \mathbf{0}_{N-M,N-M} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{G}(\mathbf{q}) \\ -\dot{\mathbf{J}}\dot{\mathbf{q}} \end{bmatrix} \tag{3.20}$$

or in a simpler form

$$\underbrace{\begin{bmatrix} \mathbf{M} & \mathbf{J}^T \\ \mathbf{J} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}(\mathbf{q})} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{D} + \mathbf{Q} \\ -\dot{\mathbf{J}}\dot{\mathbf{q}} \end{bmatrix} \tag{3.21}$$

The vector $\mathbf{D}$ contains the centrifugal, Coriolis and gravity forces and torques.

Accelerations and Lagrangian multipliers can now be calculated

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \mathbf{J}^T \\ \mathbf{J} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{D} + \mathbf{Q} \\ -\dot{\mathbf{J}}\dot{\mathbf{q}} \end{bmatrix} \tag{3.22}$$

For a forward dynamics simulation and a given set of initial conditions $\mathbf{q}(0)$ and $\dot{\mathbf{q}}(0)$, the acceleration $\ddot{\mathbf{q}}$ obtained from (3.22) can be double integrated to obtain the motion of the system in terms of the time history of the coordinates and their derivatives.

It is obvious that in order for a solution to exist, the determinant of the above square matrix $\mathbf{A}(\mathbf{q})$ (which is dependent on $\mathbf{q}$) must have full rank. Although the $\mathbf{A}$ matrix is structurally invertible for any physical system, it can be less than full rank at certain $\mathbf{q}$ values. Configurations of the manipulator, which make the above determinant zero, are called singular positions.

3.1.1.5 Constraint Forces

The Lagrangian multipliers are automatically calculated, and can be used to determine the generalised constraint forces $\mathbf{F}_c$ as:

$$\mathbf{F}_c = -\mathbf{J}^T\boldsymbol{\lambda} \tag{3.23}$$

It is also possible to calculate specific constraint force $F_c^{i,j}$ acting along q_i as a result of the *j*th constraint, as follows:

$$F_c^{i,j} = -J_{j,i}\lambda_j \tag{3.24}$$

where $J_{j,i}$, is the partial derivative of the *j*th constraint with respect to q_i, or the *j*th row and *i*th column of $\mathbf{J}$.

One of the problems with (3.21) is that the integration accumulates errors in the constraint equations. This is because the formulation only uses $\ddot{\mathbf{f}} = 0$, but not $\mathbf{f} = 0$ (for holonomic constraints) and $\dot{\mathbf{f}} = 0$. There are many different approaches proposed in the literature, but Sahinkaya stated that the following approach works well with variable step integration methods [103]. Treating the constraints as a second order system, the second block row in (3.21), *i.e.* $\ddot{\mathbf{f}} = 0$, can be replaced with

$$\mathbf{J}\ddot{\mathbf{q}} = -\dot{\mathbf{J}}\dot{\mathbf{q}} - 2\zeta\omega_n\dot{\mathbf{f}} - \omega_n^2\mathbf{f} \tag{3.25}$$

where ω_n and ζ are the effective natural frequency and damping ratio of the constraint stabilization process. A large ω_n value slows the integration, and a small value reduces the accuracy and hence the effect of the stabilization process.

In the current application, the selection of ω_n is based on the estimated suitable step size, h. This estimate can be based on the average step size used by the variable step integrator during the initial part of the simulation without stabilization. If the integration is treated as a discretization process and the average step size as the sampling interval, the Nyquist sampling theorem dictates that ω_n should be less than half of the sampling frequency. However, in practice, a factor of 5 to 10 is used instead of 2. By conveniently selecting a factor of 2π, the parameter values of $\omega_n = 1/h$ and $\zeta = 0.7$ are used. Lower ζ values cause the filter output gain exceeds one, which could destabilise the whole system. Whereas higher ζ values reduce the gains near the cut-off frequency so that more information is lost.

3.1.1.6 Inverse Dynamics

Inverse dynamics allow the calculation of required control inputs in order to achieve a desired motion. The solution of inverse dynamics relies on the forward dynamic equations written either in the form of (3.2) where $N = M$ or (3.15) where $N > M$. In the same way, the motion can be specified in terms of K independent generalised coordinates, where K is the number of degrees of the motion and can be either $K = M$ or $K < M$. Table 3.1 shows all five possible cases. Only cases A, B and C are covered here. Conventionally, a manipulator cannot perform a motion with a higher number of DOF than M. However, if a force demand is included in the motion definition and considered another DOF, it is possible for a manipulator to perform when $K > M$. This will be explained in the following subsections.

Table 3.1 Different cases for the inverse dynamics problem in terms of modelling and motion definitions

Motion / **Modelling**	$K = M$	$K < M$	$K > M$
$N = M$	A	D	-
$N > M$	B	E	C

3.1.1.7 Inverse Dynamics - Case A

This is the simplest case where the Lagrange's equations of motion are developed by using $N = M$ generalised coordinates as in (3.2), and the motion is defined by $K = N = M$ generalised coordinates.

$$\mathbf{B}\ddot{\mathbf{q}} = \mathbf{y}(t) \tag{3.26}$$

where $\mathbf{B}$ is the $K \times N$ matrix specifying the motion defining generalised coordinates, and $\mathbf{y}$ is the desired acceleration functions describing the motion. There must be K control inputs for a unique solution to exist [104].

The motion definition means that $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$, and $\ddot{\mathbf{q}}(t)$ are given. Rewriting (3.2) to include the generalised control input vector $\mathbf{u}$

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{Q} + \mathbf{u} \tag{3.27}$$

Therefore, the control input can easily be calculated as:

$$\mathbf{u} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) - \mathbf{Q} \tag{3.28}$$

Depending on the generalised coordinates, the generalised control inputs can be either torque or force, i.e. if q_i is angle then u_i is torque; if q_i is displacement then u_i is force.

It is important to note that no integration or solution of differential equations is needed for this case.

If the motion is specified by using M superfluous coordinates $\mathbf{v} = \mathbf{v}(\mathbf{q})$, then the inverse kinematics can be used to calculate the time history of the generalised coordinates, and then the first and second derivatives by using the Jacobian:

$$\dot{\mathbf{v}} = \mathbf{J}\dot{\mathbf{q}}, \qquad \ddot{\mathbf{v}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} \tag{3.29}$$

to give

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}\dot{\mathbf{v}}, \qquad \ddot{\mathbf{q}} = \mathbf{J}^{-1}\ddot{\mathbf{v}} + \mathbf{J}^{-1}\dot{\mathbf{J}}\dot{\mathbf{q}} \tag{3.30}$$

3.1.1.8 Inverse Dynamics - Case B

Case B deals with Lagrange's equations of motion using more generalised coordinates than the number of DOF of the system, *i.e.* $N > M$, as generalised by (3.21). This removes the need to eliminate redundant coordinates when writing the Lagrangian function. It also eliminates the need to perform inverse or forward kinematics when processing the simulation results, and allows the automatic calculation of constraint forces.

The motion is specified by defining $K = M$ generalised coordinates and their first and second derivatives. The $K = M$ unknown control inputs, $u_i, i = 1 \cdots M$, are added to the right hand side of the corresponding M equations in (3.20).

Then the inverse dynamics solution involves two stages:

1. Discarding the M equations corresponding to the M unknown control inputs, solve for the remaining 2(N - M) equations for the (N - M) unknown elements of $\ddot{\mathbf{q}}$ and (N - M) Lagrangian variables. Double integrating the accelerations would enable the calculation of the unspecified Lagrangian variables and their first derivatives.
2. Having calculated all $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$, the unknown control inputs $\mathbf{u}$ can be calculated by using the discarded M equations in the first stage.

3.1.2 Extended Inverse Dynamics Concept - Case C

The new approach is based on the inverse dynamics Case B. It extends the definition of motion to cover not only the acceleration of the generalised coordinates, but also the Lagrangian multipliers (or constraint forces) as follows:

$$\mathbf{B}\ddot{\mathbf{q}} + \mathbf{C}\boldsymbol{\lambda} = \mathbf{y}(t) \tag{3.31}$$

With this new definition of motion, it is now possible to specify a motion with higher DOF than that of the system, *i.e.* $K > M$. Hence, conventional inverse dynamics methods can be implemented. The following three-step formulation of the inverse dynamics solution is implemented here:

Step 1: Write the equations of motion of the system (forward dynamics) including the constraints that allow the reaction forces to be calculated by

selecting any number of suitable generalised coordinates. This is in the general form as in (3.21).

In order to demonstrate the steps mathematically, the rows in (3.21) are re-ordered so that the generalised inputs corresponding to K control inputs are grouped together (not needed in the numerical solution), *i.e.*

$$K\{\begin{bmatrix}\mathbf{M}_U & \mathbf{J}_U^T\\ \mathbf{M}_O & \mathbf{J}_O^T\\ \mathbf{J} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{q}}\\ \boldsymbol{\lambda}\end{bmatrix}=\begin{bmatrix}\mathbf{D}_U+\mathbf{Q}_U\\ \mathbf{D}_O+\mathbf{Q}_O\\ -\dot{\mathbf{J}}\dot{\mathbf{q}}\end{bmatrix}+\begin{bmatrix}\mathbf{u}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} \tag{3.32}$$

where subscript U denotes the sub-matrices or vectors containing the rows corresponding to the control inputs, and O the other rows.

Step 2: Replace the K equations involving unknown control forces $\mathbf{u}$ with the motion definitions in (3.31) to obtain:

$$K\{\begin{bmatrix}\mathbf{B} & \mathbf{C}\\ \mathbf{M}_O & \mathbf{J}_O^T\\ \mathbf{J} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\ddot{\mathbf{q}}\\ \boldsymbol{\lambda}\end{bmatrix}=\begin{bmatrix}\mathbf{y}\\ \mathbf{D}_O+\mathbf{Q}_O\\ -\dot{\mathbf{J}}\dot{\mathbf{q}}\end{bmatrix} \tag{3.33}$$

This is in the same form and size as (3.21), therefore it can be solved to obtain the motion and the Lagrangian multipliers as in the forward dynamic analysis.

Step 3: Use the discarded K equations in Step 2 to calculate the required control inputs at each integration step:

$$\mathbf{u}=\mathbf{M}_U\ddot{\mathbf{q}}+\mathbf{J}_U^T\boldsymbol{\lambda}-(\mathbf{D}_U+\mathbf{Q}_U) \tag{3.34}$$

The above three-step formulation of a general inverse dynamics analysis is valid for both conventional and the extended motion definitions, and can be applied to any system with any level of complexity as long as the equations of motion are represented in the general form as in (3.21), and the motion as in (3.31).

In order a solution to exist in Step 2, the ($2N$ - M) × ($2N$ - M) coefficient matrix in (3.33), which is a function of $\mathbf{q}$, must have a full rank. The choice of motion defining generalised coordinates (*i.e.* the structure of the $\mathbf{B}$ matrix) and the control input locations will determine the structural rank as discussed in [104].

If the motion definition does not include reaction or constraint forces, *i.e.* $\mathbf{C} = \mathbf{0}$, then defining a motion with $K > M$ would make the coefficient matrix structurally singular.

3.1.3 Numerical Example - a Constrained Two-Link Manipulator

In order to demonstrate the proposed formulation, the two-link manipulator in Figure 3.1 is utilised with the load mass being constrained on a sliding surface, as shown in Figure 3.2. The data used in the example is shown in Table 3.2. The manipulator, without the constraint on the motion of the load, has two DOF, and can be driven by two motors at the two joints. However, with the constraint on the motion of the load, the DOF of the mechanism reduces to $M = 1$. This makes one of the motors redundant. Hence a desired achievable 1- DOF load motion can be obtained by using one of the motors $K = M = 1$, and the required torque input can be calculated by using conventional inverse dynamics. However, if the motion definition also includes the specification of the ground reaction force on the load, then the proposed extended inverse dynamics analysis can be used to calculate the required torque inputs of two motors ($K = 2 > M$) uniquely to achieve the desired motion and reaction force.

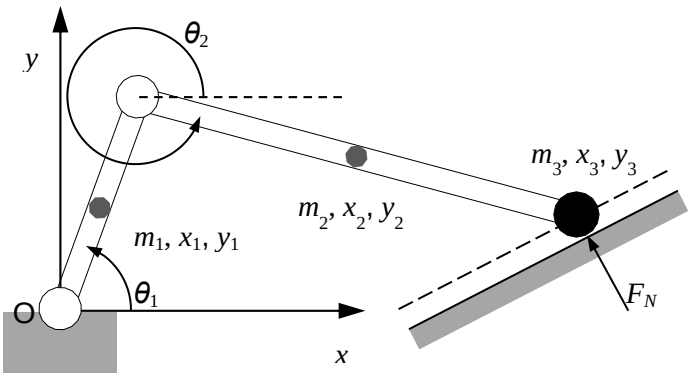

Figure 3.2 Slider-crank mechanism

Table 3.2 Data for the slider-crank mechanism in Figure 3.2

Elements		Length L_i (m)	Mass M_i (kg)	Inertia I_i (kgm^2)
1	Crank	0.5	0.5	0.01
2	Connecting rod	1.0	1.0	0.08
3	Sliding block	-	2.0	-

The sliding surface is defined by (3.35).

$$-0.3x_3 + y_3 + 0.3 = 0 \tag{3.35}$$

The desired extended motion is specified as:

$$x_3(t) = x_3(0) + \Delta x_3\left(1 - e^{-\varepsilon^3}\right) \tag{3.36}$$

$$F_N(t) = 15 \tag{3.37}$$

This represents a point-to-point smooth motion of m_3 starting at $x_3(0)$ and ending at $x_3(0) + \Delta x_3$. The normalized time variable ε is defined as:

$$\varepsilon = \frac{1.66}{T_s}t \tag{3.38}$$

where T_s is the desired 1% settling time, or duration of the motion [105].

Step 1: Ignoring the constraint on the load that causes the reaction force F_N, the equations of motion of a two-link manipulator with a load mass m_3 can be generated by using Lagrangian dynamics with or without superfluous coordinates.

If, say, four Lagrangian coordinates are selected

$$\mathbf{q} = [\theta_1, \theta_2, x_3, y_3] \tag{3.39}$$

Then two constraint equations are needed as follows:

$$\mathbf{f} = \begin{bmatrix} x_3 - L_1 cos\theta_1 - L_2 cos\theta_2 \\ y_3 - L_1 sin\theta_1 - L_2 sin\theta_2 \end{bmatrix} = \mathbf{0} \tag{3.40}$$

Now adding the third constraint equation to represent the slider mechanism that causes the reaction force F_N, will result

$$\mathbf{f} = \begin{bmatrix} x_3 - L_1 cos\theta_1 - L_2 cos\theta_2 \\ y_3 - L_1 sin\theta_1 - L_2 sin\theta_2 \\ -0.3x_3 + y_3 + 0.3 \end{bmatrix} = \mathbf{0} \tag{3.41}$$

In this formulation, the ground reaction forces on m_3 in the x-y directions can be calculated as:

$$F_c^{x_3,3} = 0.3\lambda_3 \quad and \quad F_c^{y_3,3} = -\lambda_3 \tag{3.42}$$

This gives the normal ground reaction force F_N as:

$$F_N = \sqrt{1.09}\lambda_3 \tag{3.43}$$

The constraint Jacobian matrix is

$$\mathbf{J} = \begin{bmatrix} L_1 sin\theta_1 & L_2 sin\theta_2 & 1 & 0 \\ -L_1 cos\theta_1 & -L_2 cos\theta_2 & 0 & 1 \\ 0 & 0 & -0.3 & 1 \end{bmatrix} \tag{3.44}$$

The differential algebraic equations of the motion can now be written as:

$$\begin{bmatrix} M_{1,1} & M_{1,2} & 0 & 0 & J_{1,1} & J_{2,1} & 0 \\ M_{1,2} & M_{2,2} & 0 & 0 & J_{1,2} & J_{2,2} & 0 \\ 0 & 0 & m_3 & 0 & 1 & 0 & -0.3 \\ 0 & 0 & 0 & m_3 & 0 & 1 & 1 \\ J_{1,1} & J_{1,2} & 1 & 0 & 0 & 0 & 0 \\ J_{2,1} & J_{2,2} & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -0.3 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{x}_3 \\ \ddot{y}_3 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}$$
$$= \begin{bmatrix} D_1 + Q_1 \\ D_2 + Q_2 \\ Q_3 \\ Q_4 - m_3 g \\ -L_1 cos\theta_1 \dot{\theta}_1^2 - L_2 cos\theta_2 \dot{\theta}_2^2 \\ -L_1 sin\theta_1 \dot{\theta}_1^2 - L_2 sin\theta_2 \dot{\theta}_2^2 \\ 0 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.45}$$

where the elements are defined as follows:

$$M_{1,1} = m_1 a_1^2 + I_1 + m_2 L_1^2$$
$$M_{2,2} = m_2 a_2^2 + I_2$$
$$M_{1,2} = m_2 L_1 a_2 cos(\theta_2 - \theta_1)$$
$$J_{1,1} = L_1 sin\theta_1$$
$$J_{1,2} = L_2 sin\theta_2$$
$$J_{2,1} = -L_1 cos\theta_1$$
$$J_{2,2} = -L_2 cos\theta_2$$
$$D_1 = m_2 L_1 a_2 \, sin(\theta_2 - \theta_1)\, \dot{\theta}_2^2 - (m_1 a_1 + m_2 L_1) g cos\theta_1$$
$$D_2 = -m_2 L_1 a_2 \, sin(\theta_2 - \theta_1)\, \dot{\theta}_1^2 - m_2 g a_2 cos\theta_2$$

Step 2: Remove the first two rows of (3.45) and insert the two motion specifications:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{1.09} \\ 0 & 0 & m_3 & 0 & 1 & 0 & -0.3 \\ 0 & 0 & 0 & m_3 & 0 & 1 & 1 \\ J_{1,1} & J_{1,2} & 1 & 0 & 0 & 0 & 0 \\ J_{2,1} & J_{2,2} & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -0.3 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{x}_3 \\ \ddot{y}_3 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}$$
$$= \begin{bmatrix} \Delta x_3 \left(\frac{1.66}{T_s}\right)^2 (6\varepsilon - 9\varepsilon^4) e^{-\varepsilon^3} \\ 15 \\ Q_3 \\ Q_4 - m_3 g \\ -L_1 cos\theta_1 \dot{\theta}_1^2 - L_2 cos\theta_2 \dot{\theta}_2^2 \\ -L_1 sin\theta_1 \dot{\theta}_1^2 - L_2 sin\theta_2 \dot{\theta}_2^2 \\ 0 \end{bmatrix} \quad (3.46)$$

The solution of (3.46) would give the whole motion and the Lagrangian multipliers.

Step 3: The two control inputs using the discarded two equations in Step 2 (ignoring the zero columns):

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} M_{1,1} & M_{1,2} & J_{1,1} & J_{2,1} \\ M_{1,2} & M_{2,2} & J_{1,2} & J_{2,2} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} - \begin{bmatrix} D_1 + Q_1 \\ D_2 + Q_2 \end{bmatrix} \quad (3.47)$$

If motor torques T_{m1} and T_{m2} act on the relative angles, then they can be calculated from the generalised control inputs as follows:

$$\begin{bmatrix} T_{m1} \\ T_{m2} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{3.48}$$

A typical result is shown in Figure 3.4 to Figure 3.6 for a motion of T_s = 1 s, $x_3(0)$ = 0.5 m, and Δx_3 = 0.8 m. The calculated motor torques in Figure 3.6 ensure that both the desired motion for m_3 in Figure 3.4 and the desired normal reaction force in Figure 3.5 are achieved.

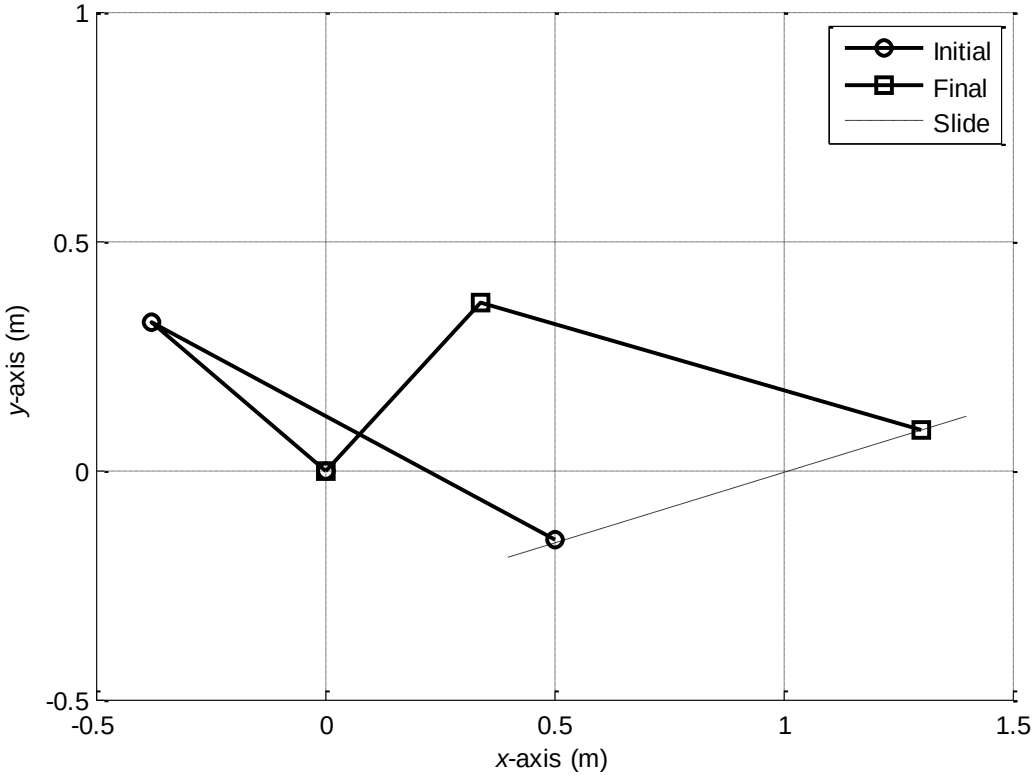

Figure 3.3 Initial and final positions of the crank mechanism in the simulation, with a sliding constraint

For comparison purposes, the results for a conventional inverse dynamics analysis for a 1- DOF desired motion are also included in Figure 3.5 and Figure 3.6, where the same desired motion of m_3 is achieved by the crank motor. Obviously, in this case there is no control on the reaction forces.

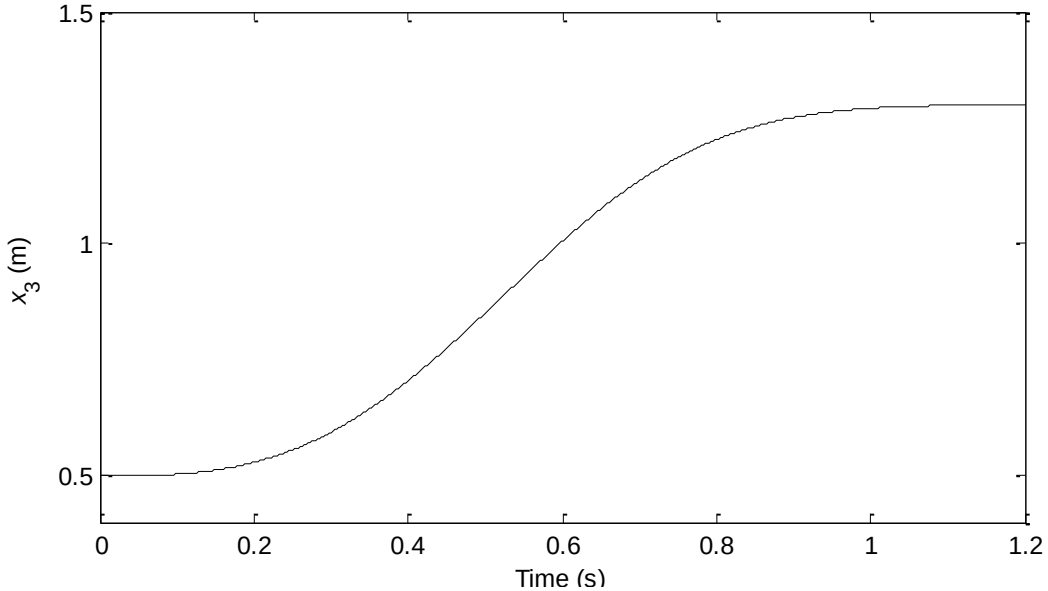

Figure 3.4 Motion definition for the simulation of a constrained crank mechanism

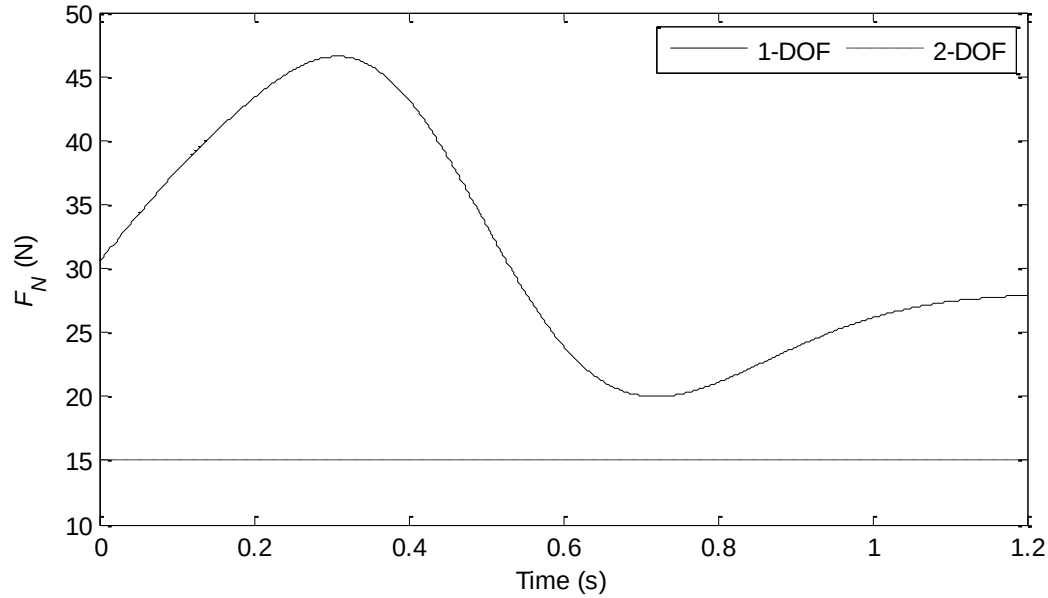

Figure 3.5 Simulation result normal reaction forces on m_3 of a constrained crank mechanism.

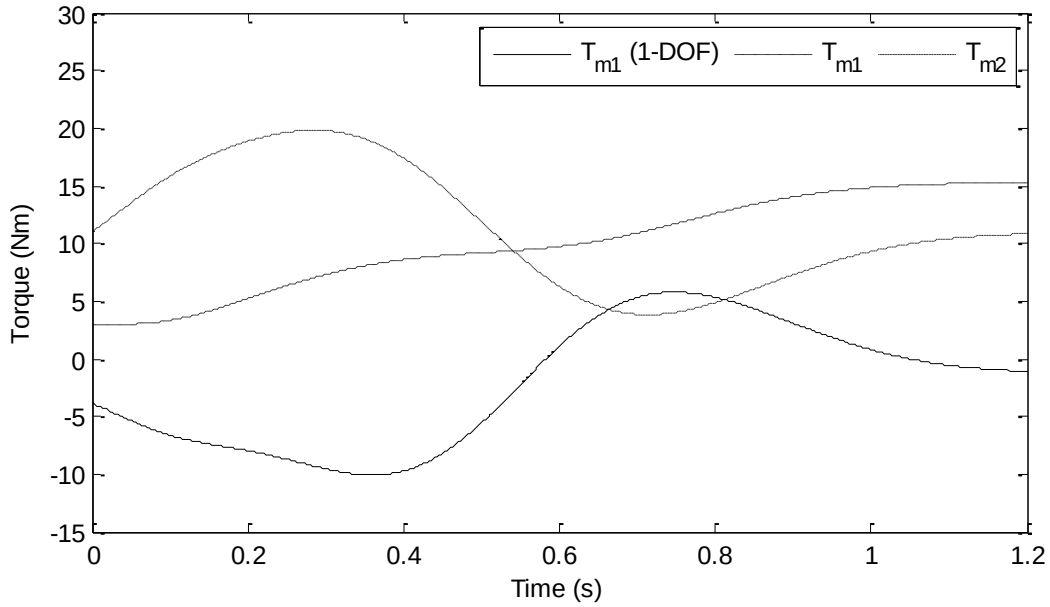

Figure 3.6 Motor torques for the simulations of a constrained crank mechanism.

3.2 Control Scheme

In this section, the controller is developed for different experimental configurations. In general, the proposed extended inverse dynamics method is used as a feedforward part of the control inputs. The controller for each joint consists of three loops: position feedback, force feedback, and inverse dynamics feedforward input, as shown in Figure 3.7.

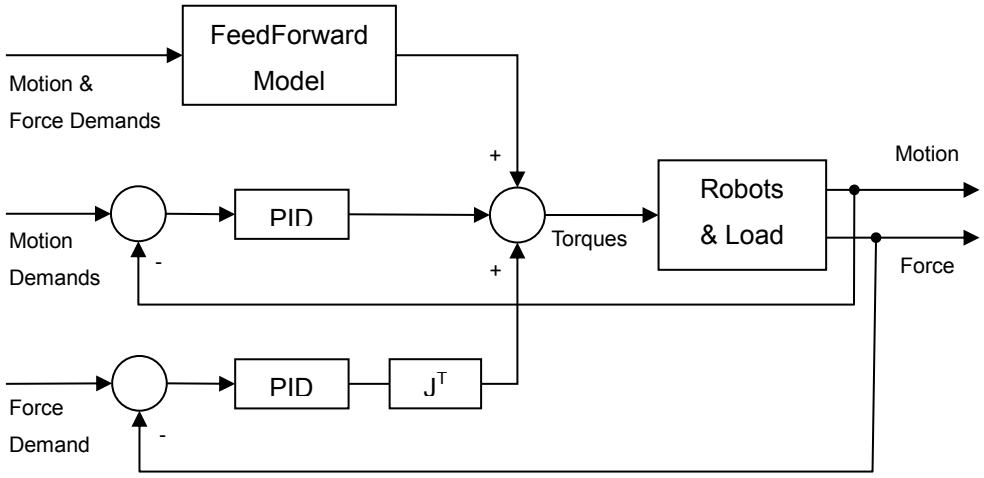

Figure 3.7 Control system diagram

$$\boldsymbol{\tau}_{Pos+Force+FMFF} = \boldsymbol{\tau}_{ForceMotionFF} + \boldsymbol{\tau}_{Pos\_Fb} + \boldsymbol{\tau}_{Force\_Fb} \tag{3.49}$$

where $\boldsymbol{\tau}_{Pos+Force+FMFF}$ is the total control input; $\boldsymbol{\tau}_{ForceFF}$ is the input from the extended inverse dynamics model feedforward loop; $\boldsymbol{\tau}_{Pos\_Fb}$ is the input from the position feedback loop; $\boldsymbol{\tau}_{Force\_Fb}$ is the input from the force feedback loop. This controller is called Controller 1 and noted as “Pos + Force + FMFF”.

For comparison, different controllers are introduced in the experiments. A conventional motion-only inverse dynamics model is built [102].

$$\boldsymbol{\tau}_{Pos+Force+MFF} = \boldsymbol{\tau}_{MotionFF} + \boldsymbol{\tau}_{Pos\_Fb} + \boldsymbol{\tau}_{Force\_Fb} \tag{3.50}$$

where $\boldsymbol{\tau}_{MotionFF}$ is the input from the conventional motion-only inverse dynamics model feedforward loop. This controller is called Controller 2 and noted as “Pos + Force + MFF”. It represents conventional inverse dynamics controller.

$$\boldsymbol{\tau}_{Pos+Force} = \boldsymbol{\tau}_{Pos\_Fb} + \boldsymbol{\tau}_{Force\_Fb} \tag{3.51}$$

This controller is called Controller 3 and noted as “Pos + Force”. It represents the conventional hybrid position/force controller.

$$\boldsymbol{\tau}_{Pos+FMFF} = \boldsymbol{\tau}_{ForceMotionFF} + \boldsymbol{\tau}_{Pos\_Fb} \tag{3.52}$$

This controller is called Controller 4 and noted as “Pos + FMFF”.

$$\boldsymbol{\tau}_{Pos} = \boldsymbol{\tau}_{Pos\_Fb} \tag{3.53}$$

This controller is called Controller 5 and noted as “Pos Only”.

Controllers 1-5 represented by (3.49) - (3.53) are used to carry out the experiments and performances are compared in the later sections.

3.3 Application to Multi-robot Cooperative Systems

In this section, the above-proposed controller is built and simulated for an example cooperative system. This example is a simple version of the experimental configuration Case II as described in Section 6.1. In this simulation,

the two Omni robots are simplified to two two-DOF manipulators moving on a vertical plane, where a load object is being held between the manipulators, as shown in Figure 3.8. The simulated system is presented in the following subsections.

3.3.1 System Description

The two identical manipulators are used to perform a cooperative handling job. There are in total four actuators in the system, *i.e.* one electric motor at each joint. Figure 3.8 shows the simulated mechanism. The positive direction of each motor angle and torque is shown in the figure. The two manipulators are located 2.2 m apart from each other, and each link and motor is numbered as shown. They are both connected to the load from their end-effectors, *i.e.* Link 1-2 and Link 2-2. At the initial position, the centre of the load is located at point (0.8 0) and its orientation is parallel to the x-axis, *i.e.* θ_5 = 0°. The parameters are listed in Table 3.3.

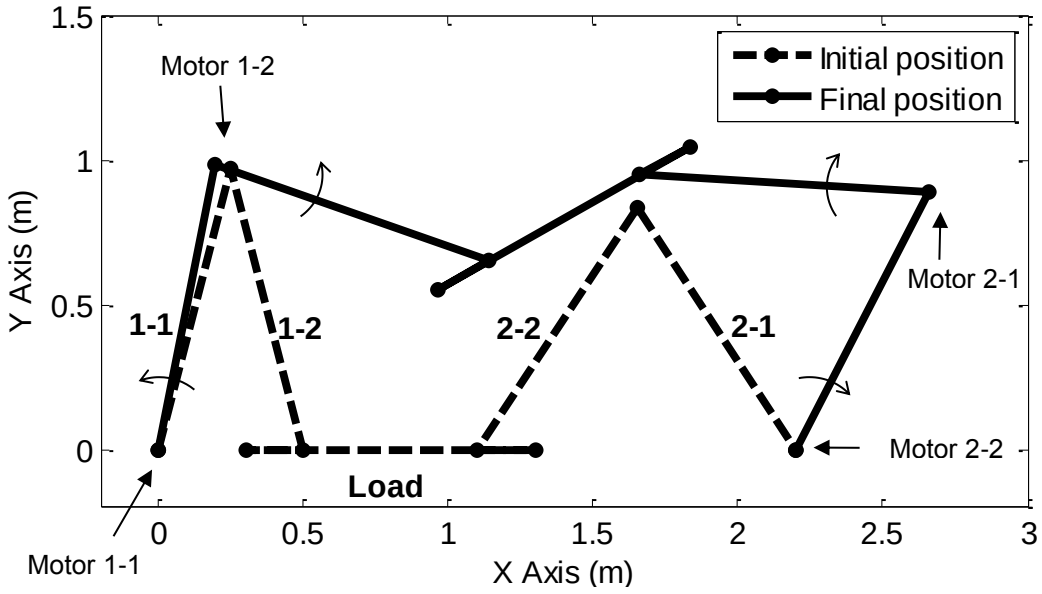

Figure 3.8 Initial position and final position

Table 3.3 System parameters

	Link 1-1	**Link 1-2**	**Link 2-1**	**Link 2-2**	**Load (Link 3)**
Mass (kg)	1	1	1	1	2
Length (m)	1	1	1	1	0.6
Inertia (kgm^2)	0.08333	0.08333	0.08333	0.08333	0.1667

In the simulation, the forward and inverse modelling of the system is provided by Dysim (Section 4.1.2.3) blocks. The variable step ode45 (Dormand-Prince) solver is selected, with a relative tolerance setting of 10^{-6}. The whole simulation runs for five seconds.

In the simulation, a straight-line point-to-point motion is given as the desired motion. The following path equations will specify a motion with zero initial and final velocity and acceleration, and it is continuous for all $t \geq 0$ [106].

$$x_L = x_L(0) + \Delta x_L\left(1 - e^{-\varepsilon^3}\right) \tag{3.54}$$

$$u = \alpha t \tag{3.55}$$

Δx_L denotes the required change in x_L, and ε is the normalized time. The time scaling parameter α determines the speed of the motion and relates to the 1% settling time, T_s, as follows:

$$\mathrm{T_s} = \frac{1.66}{\alpha} \tag{3.56}$$

The path equation also has continuous first and second derivatives:

$$\dot{x}_L = 3\Delta x_L \varepsilon^2 e^{-\varepsilon^3} \tag{3.57}$$

$$\ddot{x}_L = 3\Delta x_L (2\varepsilon - 3\varepsilon^4) e^{-\varepsilon^3} \tag{3.58}$$

A unit point-to-point profile is shown in Figure 3.9, *i.e.* when $x_L(0) = 0$ m, $\Delta x_L = 1$ m, and $\alpha = 1.66$ thus $T_s = 1$ s.

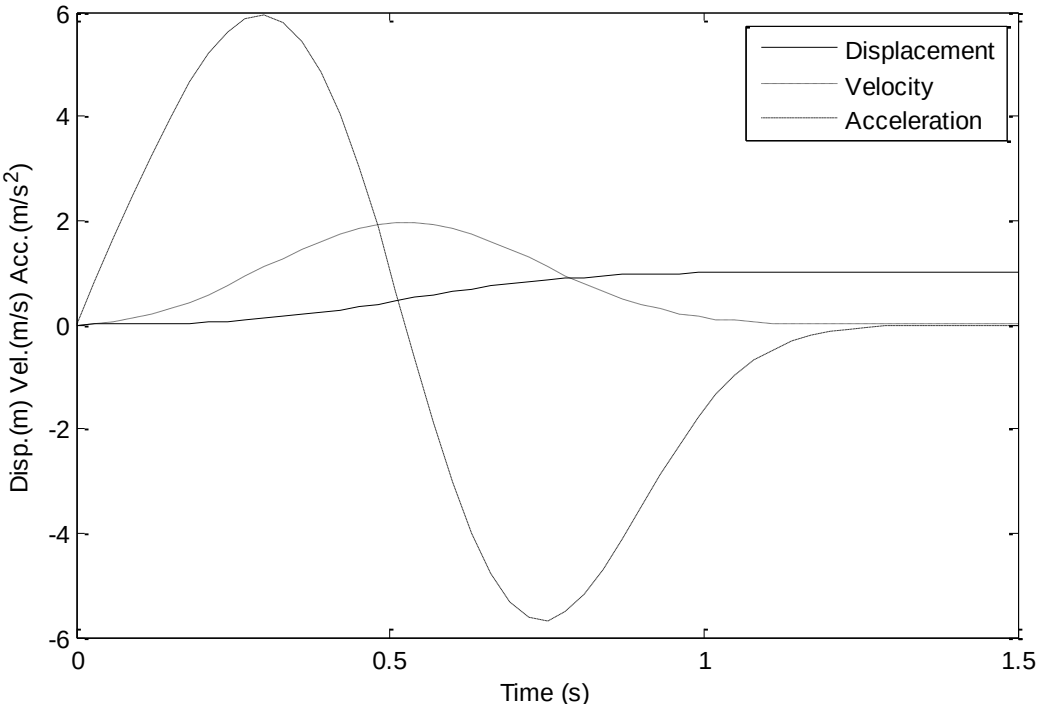

Figure 3.9 Point-to-point motion when $x_L(0)$ = 0 m, Δx_L = 1 m, and α = 1.66

The same desired motion equations are used for y_L with ΔY_L and θ_5 with $\Delta\theta_5$ denoting the required change in y_L and θ_5. The initial position and orientation of the load is $x_L(0)$ = 0.8 m, $y_L(0)$ = 0 m and $\theta_5(0)$ = 0°. It is desired to move to the destination at $x_L(T_s)$ = 1.6 m, $y_L(T_s)$ = 0.8 m and $\theta_5(T_s)$ = 30° on a straight line. The 1% settling time T_s and simulation period T are selected as 3 s and 5 s, respectively, so that the motion lasts 3 s with 1 s before and after the motion.

3.3.2 Simulation with Conventional Hybrid Position/Force Control

Since the motion requires only three DOF, there is one redundant actuator in the system. Thus, this actuator can be used to control the force which the manipulators apply to the load, *i.e.* compression or tension force. This is one of the simplest forms of the hybrid force/position control [51].

In the simulation, three motors control the linear and rotational movements of the load, which leaves one motor to be in charge of controlling the reaction force. To choose the force controlling actuator in this motion, test runs are carried out on each motor. The inverse dynamics fails if Motor 1-1 or Motor 2-1 is chosen as the redundant one, because the desired motion is physically impossible to be realized using the other three motors. Hence, one of the upper joint motors is

selected for the task. Figure 3.10 shows the motor torques when Motor 1-2 is chosen and set to a constant torque value.

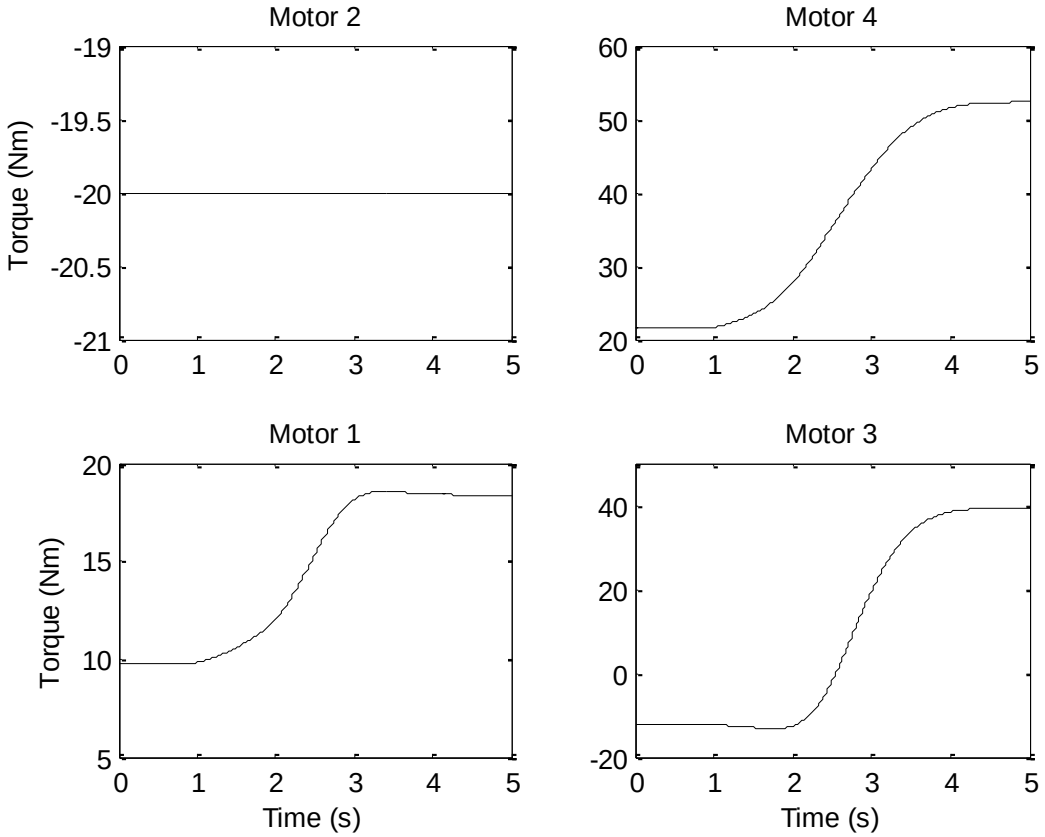

Figure 3.10 Motor torques when Motor 2 is selected as the force controlling motor and given a constant torque

Forces in generalised coordinates can be calculated by the Dysim Simulink block (Section 4.1.2.3) using equation (3.24), which simulates means of force sensing. In particular, the forces between each upper link and the load are computed. In terms of the reaction forces, the forces in generalised coordinates calculated by Dysim forward dynamics have to be transformed into the body-fixed coordinate of the load object in order to simulate the reaction forces.

The generalised forces calculated by the forward system block are the forces in earth-fixed coordinate directions, one in horizontal x direction and the other in vertical y direction. To transform the forces to the load coordinates, the angular displacement of the load is needed.

$$\begin{bmatrix} F_{xLoad} \\ F_{yLoad} \end{bmatrix} = \begin{bmatrix} cos\theta_5 & sin\theta_5 \\ -sin\theta_5 & cos\theta_5 \end{bmatrix} \begin{bmatrix} F_{x_5} \\ F_{y_5} \end{bmatrix} \tag{3.59}$$

where F_{xLoad} and F_{yLoad} are the reaction forces that Link 1-2 and Link 2-2 exerted on the load in the load's body-fixed coordinates. Figure 3.11 shows the load coordinates.

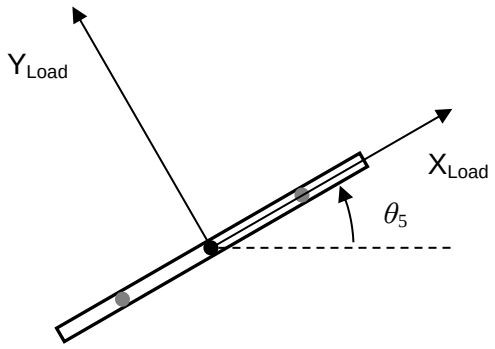

Figure 3.11 Load coordinates

A block diagram of the control system is illustrated in Figure 3.7. Combined with the force controlling torque, the inverse dynamics model of the system is able to generate three torques for Motors 1-1, 2-1 and 2-2, which perform the motion. The forward model of the system then calculates the motion and the internal forces which both upper links exert on the load. The forces are then transformed into the load's coordinates in order to perform the feedback control.

The system passes a singular configuration in this motion. The singular point is reached when the load and the upper link of the second manipulator turn into a straight line. The determinant of the inverse dynamics (the matrix on the left-hand side of (3.33)) [104] indicates the singular point as shown in Figure 3.12. To avoid the simulation collapsing, the minimum step size is set to 0.01 in the configuration parameters. In a real system, the mechanism is able to pass the singularity point due to the momentum of the load and links.

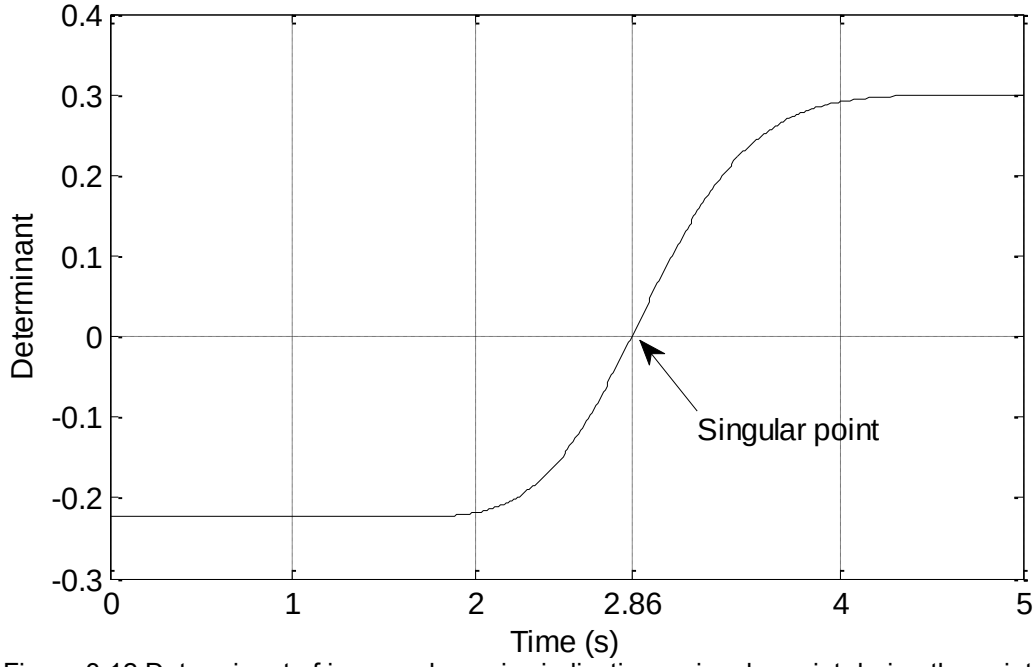

Figure 3.12 Determinant of inverse dynamics indicating a singular point during the point-to-point motion

At first, without any force feedback, a constant torque of -20 Nm is given to the force controlling motor while the other three are controlling the motion. Figure 3.13 illustrates the reaction forces in the load x direction. It is noted that the directions of the forces are changing during the motion. From the load's point of view, the forces exerted by both links change from squeezing to pulling during the motion. The force from Link 1-2 remains in the same direction throughout the motion - pushing the load, whereas the force from Link 2-2 changes direction at the singular point - from pushing the load to pulling. The combined force of around 10N from Link 1-2 and Link 2-2 at the end position is evidence for part of the weight of the load.

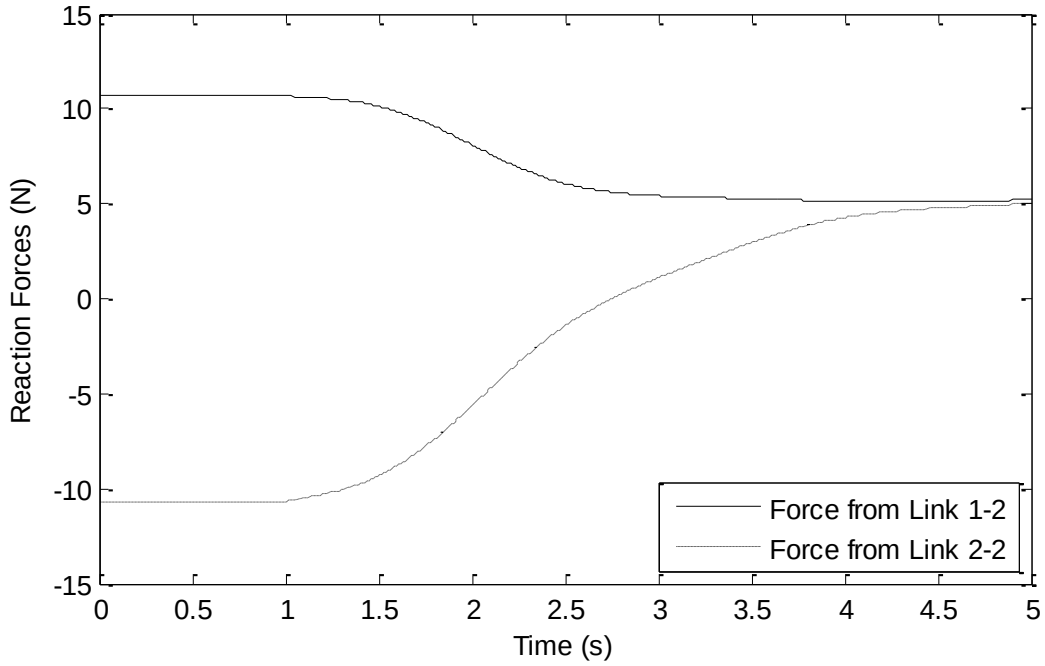

Figure 3.13 Reaction forces in X_{Load} direction using constant torque on the force controlling motor

However, if a desired reaction force is specified, then a closed-loop force feedback controller is necessary to achieve the goal. The desired reaction force for Link 1-2 is set to - 20 N, *i.e.* a compression force on the load. A PI controller is introduced to regulate the motor torque according to the force error. The proportional gain and integration gain are both set to 100 after trial and error.

Figure 3.15 shows the reaction forces detected in the cooperative motion in load coordinates. It is noticed that the two reaction forces from the two manipulators are not equal. The reason is that the weight of the load and the forces from the motion have not been excluded from the total reaction forces. The zoomed in result in Figure 3.16 shows that the force error from Link 1-2 is within ± 0.2 N, *i.e.* ±1% of the desired force.

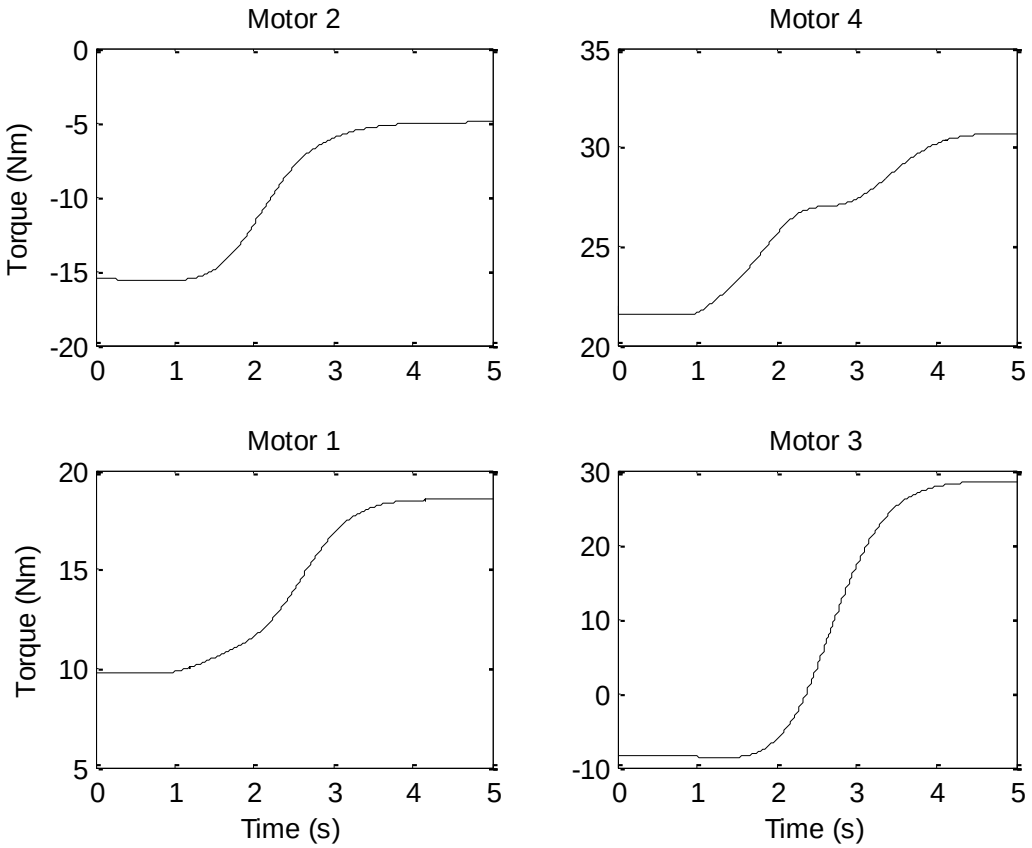

Figure 3.14 Motor torques when Motor 2 is used as the force controlling motor

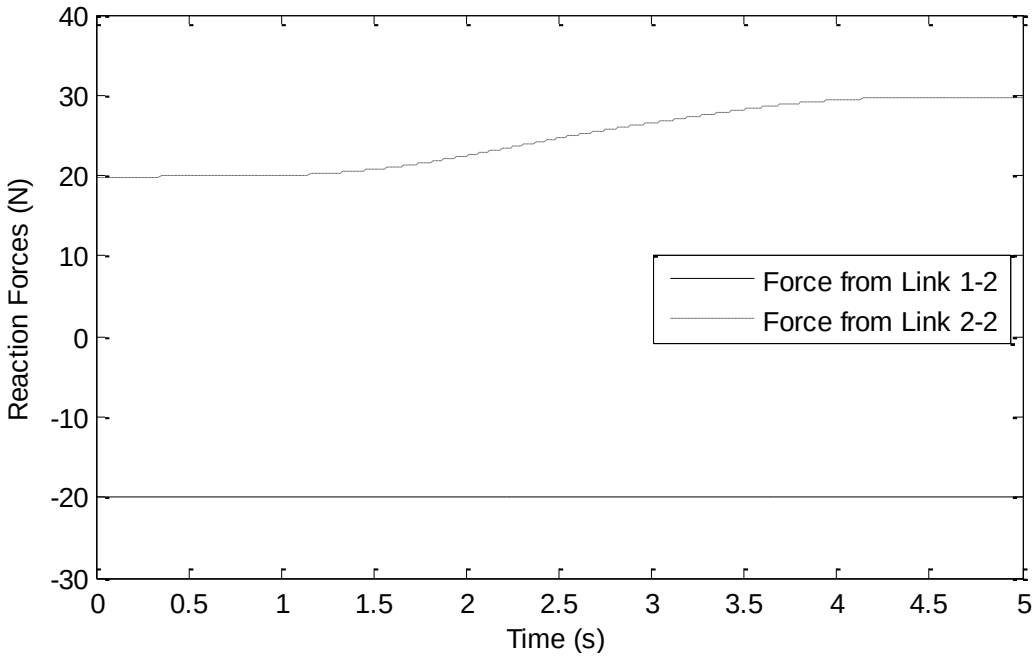

Figure 3.15 Reaction forces on the load (F_{xLoad}) using feedback control on the force controlling motor

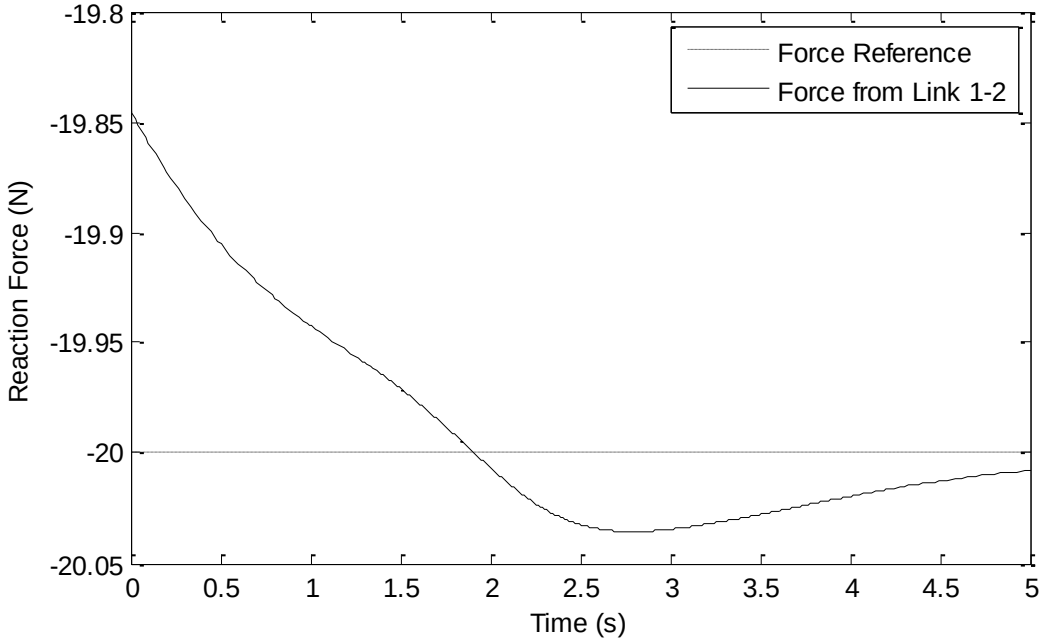

Figure 3.16 (Zoom in on Figure 3.15) Reaction force on the load (F_{xLoad}) from Link 1-2 with closed loop control

3.3.3 Simulation with the Extended Inverse Dynamics Feedforward Inputs

With the same point-to-point motion and force demand, simulation is carried out with the extended inverse dynamics feedforward input. The difference now is that all the motor torques are calculated by the extended inverse dynamics (3.34), combining both motion and force torques.

Figure 3.17 shows the motor torques calculated using the extended inverse dynamics model. It is noted that the torques are almost the same as those from the hybrid position/force control, in Figure 3.14. This is because the motion and force demands are the same therefore the torques needed to achieve such motion and force should be the same regardless of how they are obtained, whether recorded from the hybrid controller or calculated by the extended inverse dynamics. Similar to Figure 3.15, Figure 3.18 shows the reaction forces acting on the load by the links.

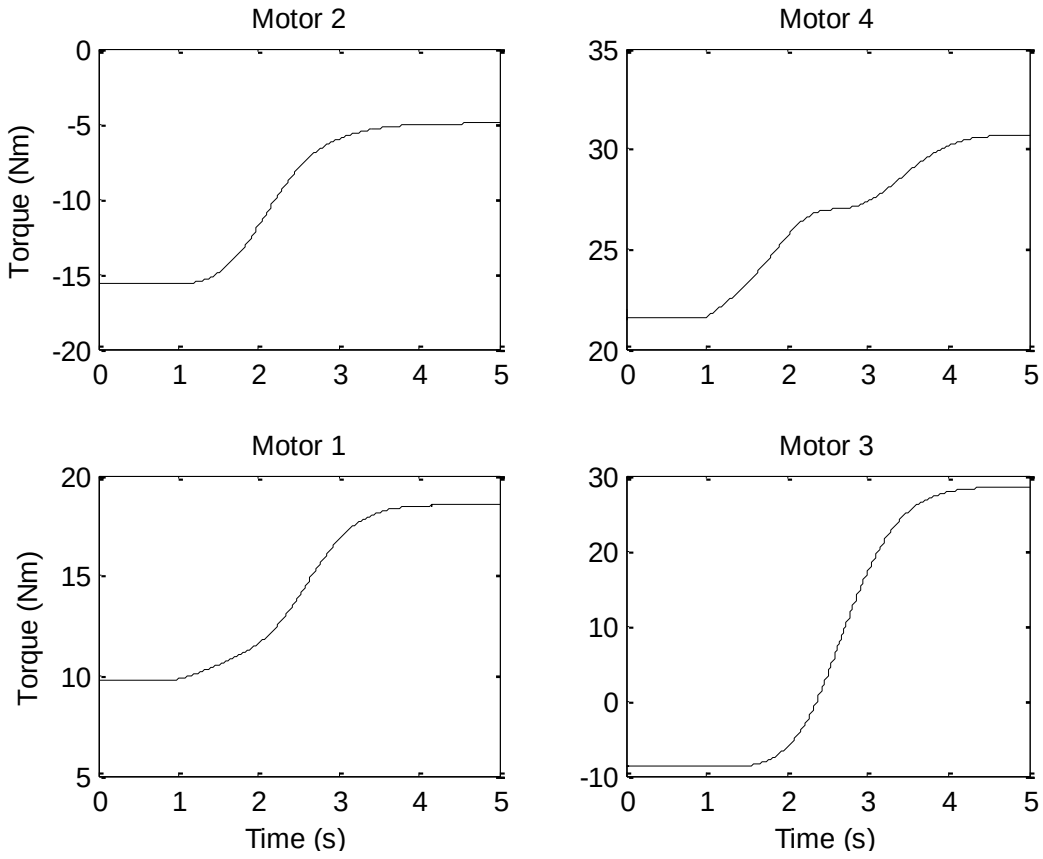

Figure 3.17 Motor torques for the simulation calculated using the extended inverse dynamics controller

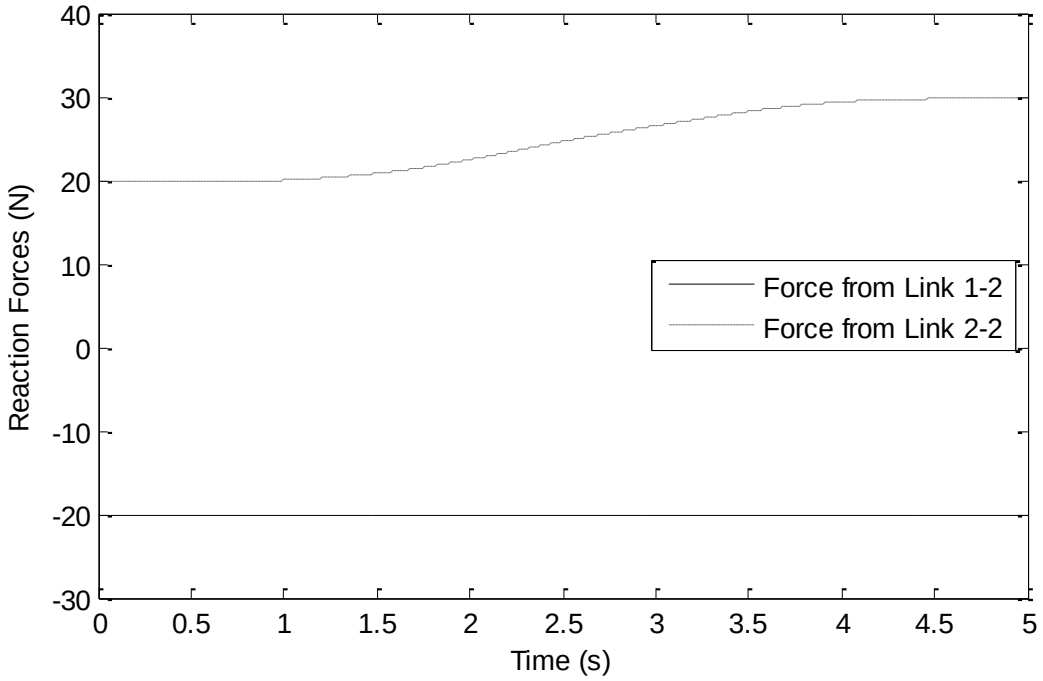

Figure 3.18 Reaction forces on the load (F_{xLoad}) using the extended inverse dynamics controller

3.4 Concluding Remarks

This chapter introduced a new formulation to calculate the inverse dynamics of a system with both motion and force demands.

The inverse dynamics modelling is based on the constrained Lagrangian formulation. The use of redundant coordinates allows different coordinates to be used to specify the trajectory, control inputs, motion measurement, and disturbances. The idea of extended definition of motion allows the calculation to be performed with conventional inverse dynamics analysis.

A cooperative control system of two 2- DOF manipulators has been simulated. The two manipulators are carrying a common load trying to complete a given load motion with desired internal force. A control strategy for both motion and force tracking of the non-linear system based on the extended inverse dynamics controller is examined.

The following chapter introduces the experimental set-up and modelling of the robot involved. Then a new approach for finding optimal excitation trajectories for the purpose of dynamic parameter estimation was explained and demonstrated.

4 Experimental Set-up and Dynamic Modelling

The previous chapter introduced a new approach to calculate the inverse dynamics of redundant manipulator systems, taking into account of both force and motion inputs. A controller was then built using the new inverse dynamics model as a feedforward path.

In this chapter, the experimental set-up and modelling of the robot systems is performed. The first half of this chapter presents the hardware and software involved and two experimental configurations which will be used in experiments in Chapter 5 and 6. The second half of the chapter presents a new approach toward the design of optimal robot excitation trajectories and its application to parameter estimation for dynamic modelling. Briefly, the method uses the fundamental frequency, number of harmonics and overall gain as the parameters of a Schroeder Phased Harmonic Sequence (SPHS) signal to define optimised excitation trajectories. The results of this approach are also published in [107].

The chapter is organised as follows: Section 4.1 presents all the hardware and software that is used later in the simulations and experiments; Section 4.2 presents different configurations for the experiments; Section 4.3 presents the dynamic modelling of the robot. Section 4.4 concludes this chapter.

4.1 Hardware and Software

The system is implemented on the Geomagic® Touch™ Haptic Device (formerly SensAble Phantom Omni, hereafter referred to as the “Omni robot” or “Robot”) by using a “soft real-time” control interface with Simulink developed by Quanser [108, 109].

4.1.1 Hardware Description

4.1.1.1 Geomagic Touch Haptic Device and Custom Made Stands

The Omni robot, shown in Figure 4.1, is originally an input device that has some force feedback capability. In this thesis it is used as a robot manipulator, utilising the motors fitted originally to provide force feedback.

The Omni robot is a force feedback haptic device, it has relatively low gear ratios (approximately 1:8 for Joint 0, 1 and 2) for exerting sensible forces, which makes it a good experiment subject for the extended inverse dynamics controller.

The Robot has six revolute joints and the configuration is similar to a typical serial industrial manipulator. However, among all six joints only the first three are actuated by DC electric motors; the other three are free rotating wrist joints. The three actuated joints are shown as J0, J1 and J2.

The inputs to the robot are Pulse Width Modulation (PWM) indices, which control the DC voltage across the motor armatures. The outputs of the robot are joint angles which come from two types of sources: digital encoders and potentiometers. The three encoders on motors 0, 1 and 2 output digital readings directly through the interface. The three potentiometers in the wrist joints output analogue voltages which in turn are converted by analogue to digital converters (ADCs) into digital readings. All readings are then translated into practical units to indicate the angle for each joint.

Due to the lack of detailed information from the manufacturer's specification document, experimental parameter estimation is required. Parameters such as link mass, moment of inertia, and various friction coefficients will need to be identified. Table 4.1 lists some relevant specifications from the document provided by the manufacturer [108].

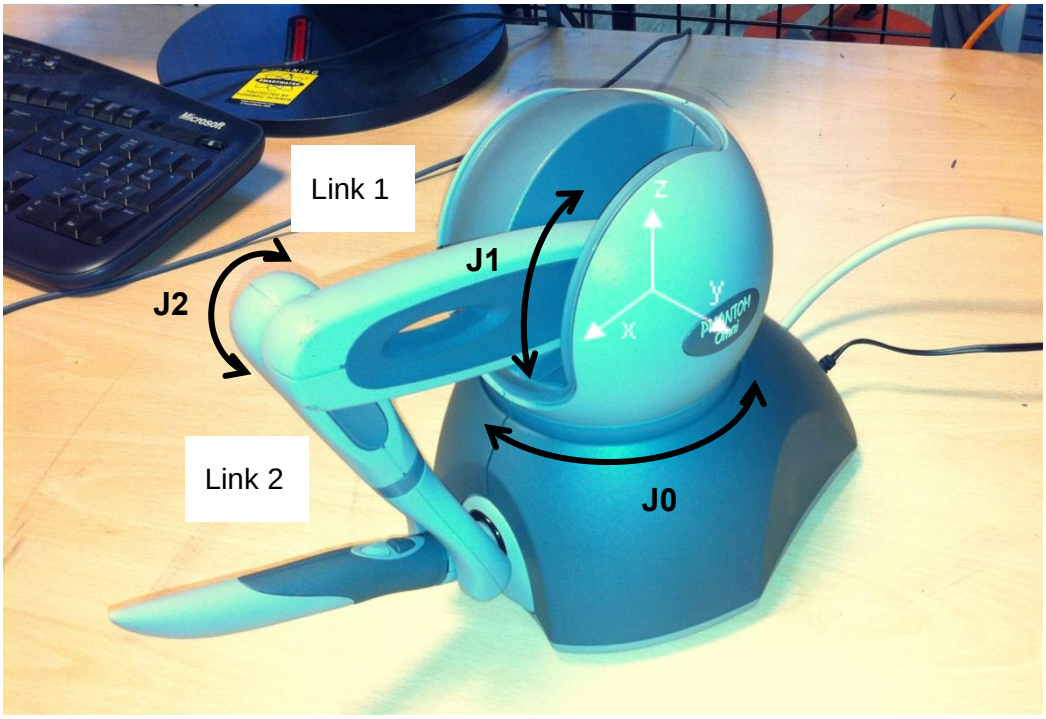

Figure 4.1 Geomagic Touch Haptic Device (formerly SensAble Phantom Omni) with joint and link labels

Table 4.1 The PHANTOM Omni Device specification (partial)

Backdrive friction	< 1 oz (0.26 N)
Maximum exertable force at nominal (orthogonal arms) position	0.75 lbf. (3.3 N)
Continuous exertable force (24 hrs.)	> 0.2 lbf. (0.88 N)
Stiffness	X axis > 7.3 lb/in (1.26 N/mm) Y axis > 13.4 lb/in (2.31 N/mm) Z axis > 5.9 lb/in (1.02 N/mm)
Inertia (apparent mass at tip)	~0.101 lbm. (45 g)
Position sensing **[Stylus gimbal]**	x, y, z (digital encoders) [Pitch, roll, yaw (±5% linearity potentiometers)]

To avoid unwanted movements during experiments, the Omni robot has to be securely mounted. There are no readily available parts, therefore a stand is designed and manufactured. The design is in Solid Edge, as shown in Figure 4.2. Each stand is made from aluminium plate, aluminium columns, acrylic plate and medium-density fibreboard. It is then made and assembled. A holding fixture to lock Link 2 is also designed and made, it is part of an experiment

design to facilitate a physical constraint, more details can be found in Sections 4.2, 5.1, and 6.1.

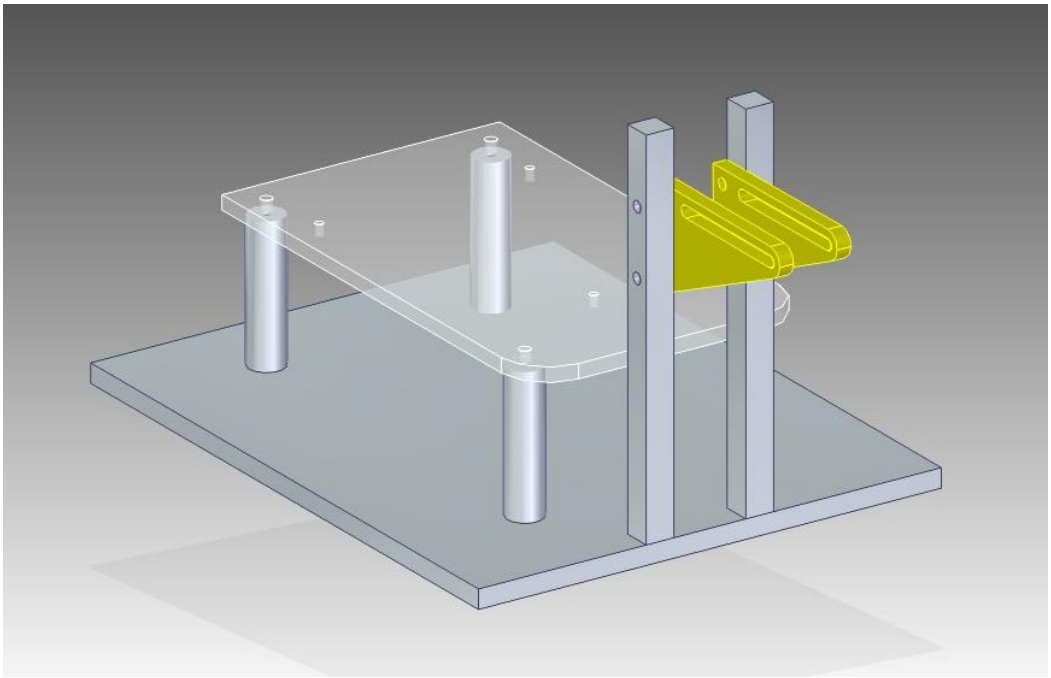

Figure 4.2 CAD assembly of the robot stand and a holding fixture for Link 2

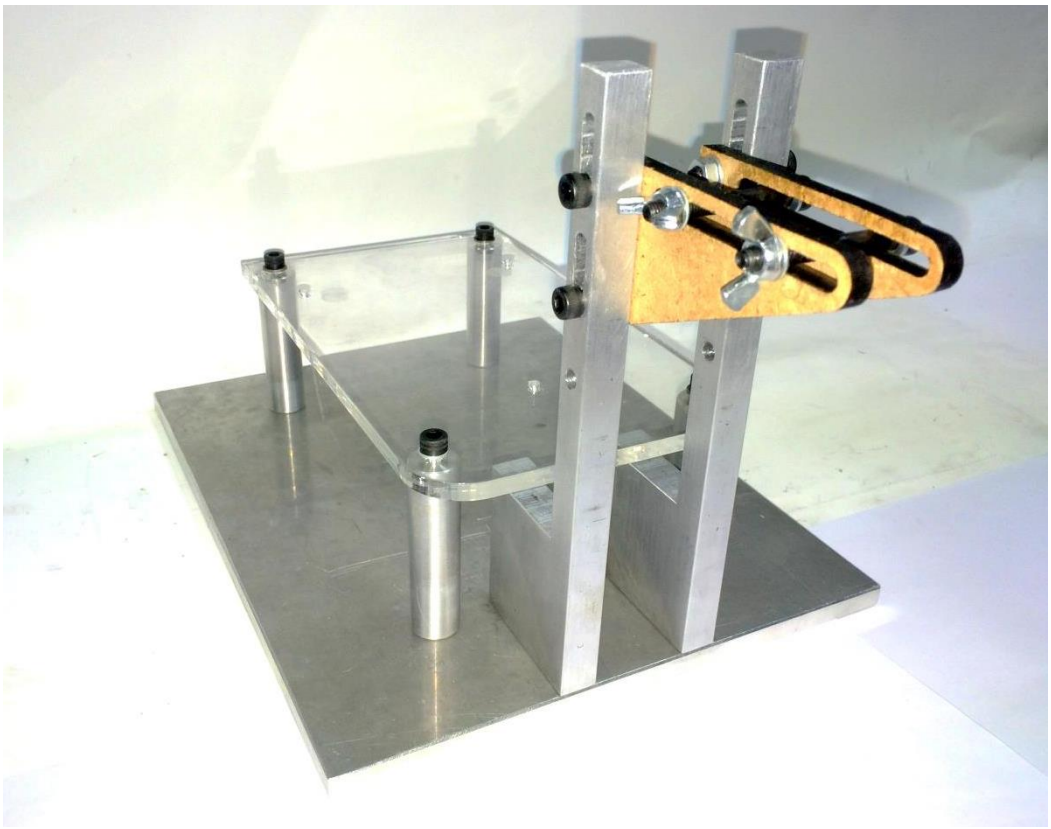

Figure 4.3 The actual robot stand and the holding fixture for Link 2

The robots are secured on the stands with screws, as shown in Figure 4.4. The stands are then clamped down on the table so that the robots are fixed without any relative movement.

Figure 4.4 One of the Omni robots on the stand

4.1.1.2 Design, Production and Installation of the Load Assembly

As shown in Figure 4.1, the Omni robot has a pen-shaped end-effector, also known as a stylus, at the end of Link 2. It is designed for easy manoeuvre by hands. However, this makes it difficult to be connected to other objects, such as a force sensor or another Omni robot. In order to use the Robot in experiments, a rigid connection has to be made either between one robot and its environment or between two robots. Because there are no ready-made parts for the job, a connection part has to be made, which also acts as part of the load of the system.

According to the product manual, the upper half of the stylus can be taken off and exchanged for service purposes. The connecting mechanism between the upper and lower halves is a 6.35 mm jack socket and plug, respectively. After removing the upper half, the jack plug, as shown in Figure 4.4, can be used as a

connection point for the custom-made part. This configuration also utilises the wrist joints as a free rotating load gripping point.

The connection part is also designed in Solid Edge. Then it is produced from a 3D printing machine. The CAD drawing and printed part are shown in Figure 4.5. It is designed to utilise the jack plug on the lower half of the stylus and it is secured by bolts.

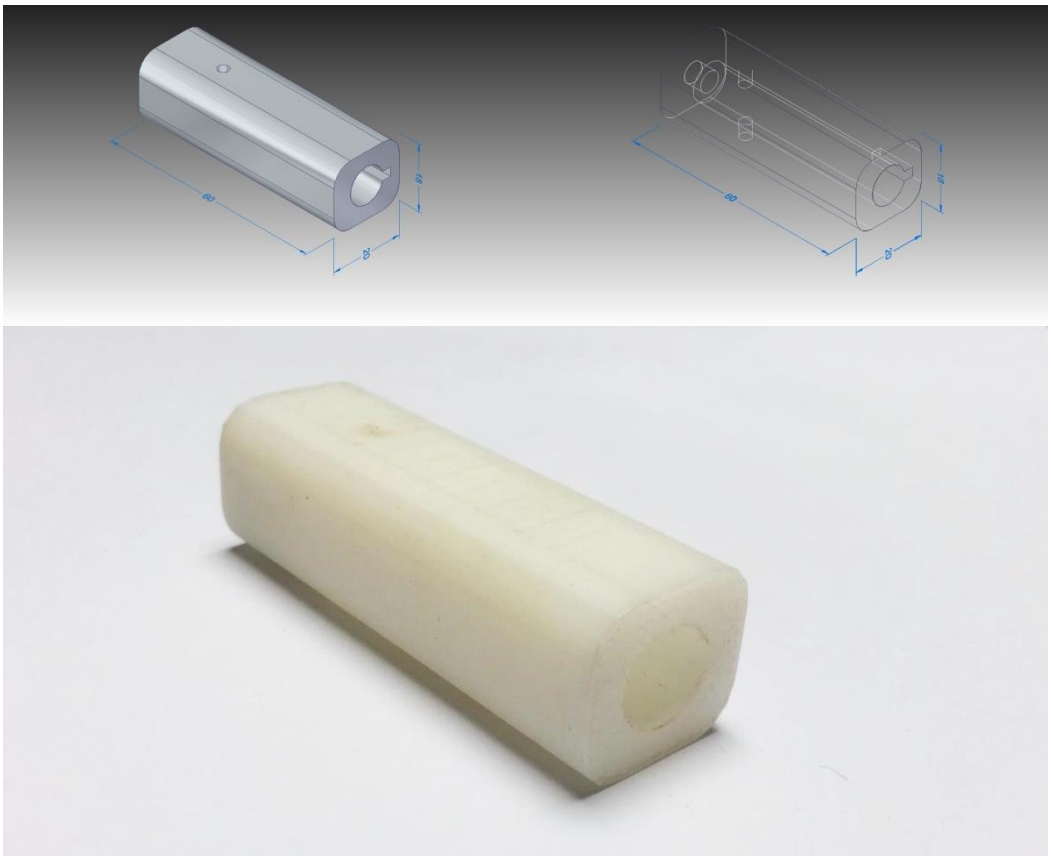

Figure 4.5 Load part: CAD model and 3D printed part

The part is also designed to accommodate a force sensor in the middle. The two separate halves are bolted to the force sensor; and together they form the load for the experiments. Here it is assumed that the connections between the load and two end-effectors can rotate without any friction as the load is being moved; and the centre of gravity of the load is in the middle.

4.1.1.3 Force Sensor Installation and Calibration

Force and motion control of robot manipulators is in the core of all robotic applications concerning robots and environment interactions. An increasing number of applications require accurate force manoeuvrability among

manipulators [110]. In order to have successful collaborative robot control, control of the interaction force is essential. With the ability to measure reaction force or forces, for example, redundant actuators can be utilised to regulate the force in a feedback loop while motion is executed by the other actuators.

In this thesis, force measurement is required for both force feedback control and to verify the control performance. The force measurement between the robot and its environment and between two robots is accomplished by a miniature load cell mounted in the middle of the link (Figure 4.6). The load cell used here is the one-axis miniature 'S' beam DBBSMM series made by Applied Measurements Limited [111]. The model chosen for the experiment has a 2 kg capacity or approximately 20 N in both directions along its working axis, *i.e.* tension and compression. The 'S' beam model is chosen because it is able to withstand lateral force. It is used to measure the internal force of the load alone its primary axis.

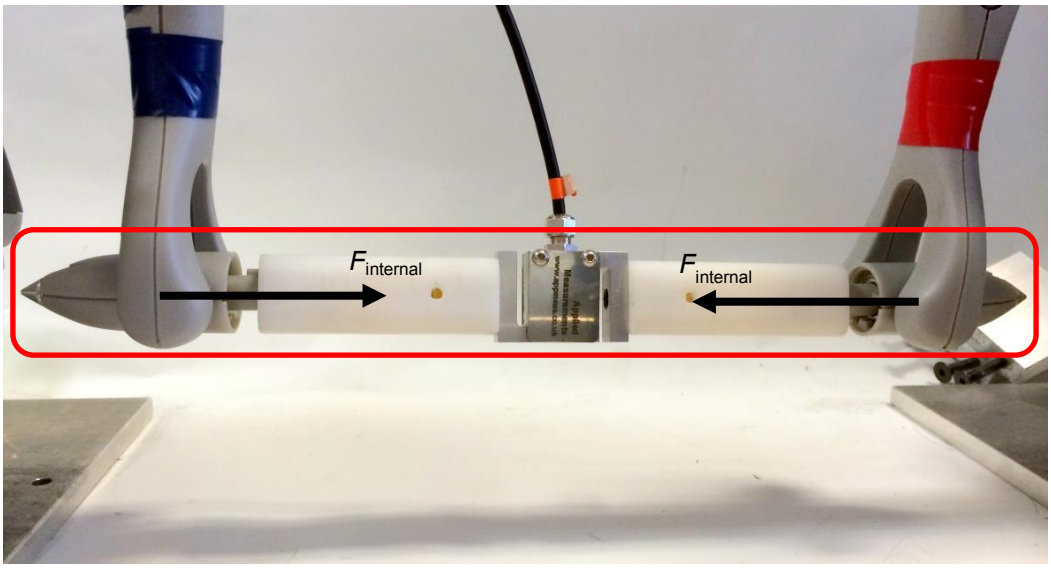

Figure 4.6 Force sensor mounted in the middle of the load, which together with the rest of the 'pen' from both robots, forms the load.

The input voltage of the force sensor is 10 volts. The rated output of the sensor is 10 mV/kg. The output is then amplified by a conditioning circuit to improve the resolution of the ADC. Although the maximum static force the Omni robot can exert is 3.3 N [108], the dynamical force can be larger when the robot is moving fast. Hence the range of the force sensor is set to 10 N in both directions, *i.e.* to adjust the conditioning circuit to amplify the force sensor output of ±10 mV by a

factor so that the output fits the input range of the ADC, maximising the resolution. The input to ADC channels has a range of 0 to 3.2 volts.

The data acquisition for the force sensor is achieved by using one of the Omni robot's existing ADC channels. As mentioned in Section 4.1.1.1, potentiometers are used at the wrist joints of the Omni robot to measure angular displacement. The voltages are then fed into the ADCs inside the robot. Given that the angle of the last joint, to twist the stylus, is not needed in the experiments (planar experiments only), hence the connection to the ADC channel that corresponds to the last joint is disconnected from its potentiometer to make use the existing channel. The newly available channel is then connected to the force sensor amplifier output. This allows the ADC to convert the amplified force reading into digital form. It also means that measurements are synchronised by the robot's hardware. The interpretation of this measurement will be force instead of angle.

The advantage of using existing components is that no extra hardware, such as ADC cards, is needed. Furthermore, the entire system is easier to implement, because it uses the same software interface as the robot. This significantly reduces the effort and time to integrate separate hardware and software systems.

To calibrate the force sensor, a 100-gram weight set is used to exert both tension and compression forces. As shown in Figure 4.7, the weights are added in 100-gram increments. Then the ADC outputs are recorded three times for each increment and average values are obtained. The graph shows that the force sensor reading behaves linearly and the relation between force and ADC reading can be extracted, resulting in a constant, C_{force}. The value of the constant, -1.8515, can be read out from the graph and this value will be used in all the experiments in this thesis.

$$F = \frac{g}{-1.8515} b \tag{4.1}$$

where f is force; g is the acceleration of gravity; b is ADC reading.

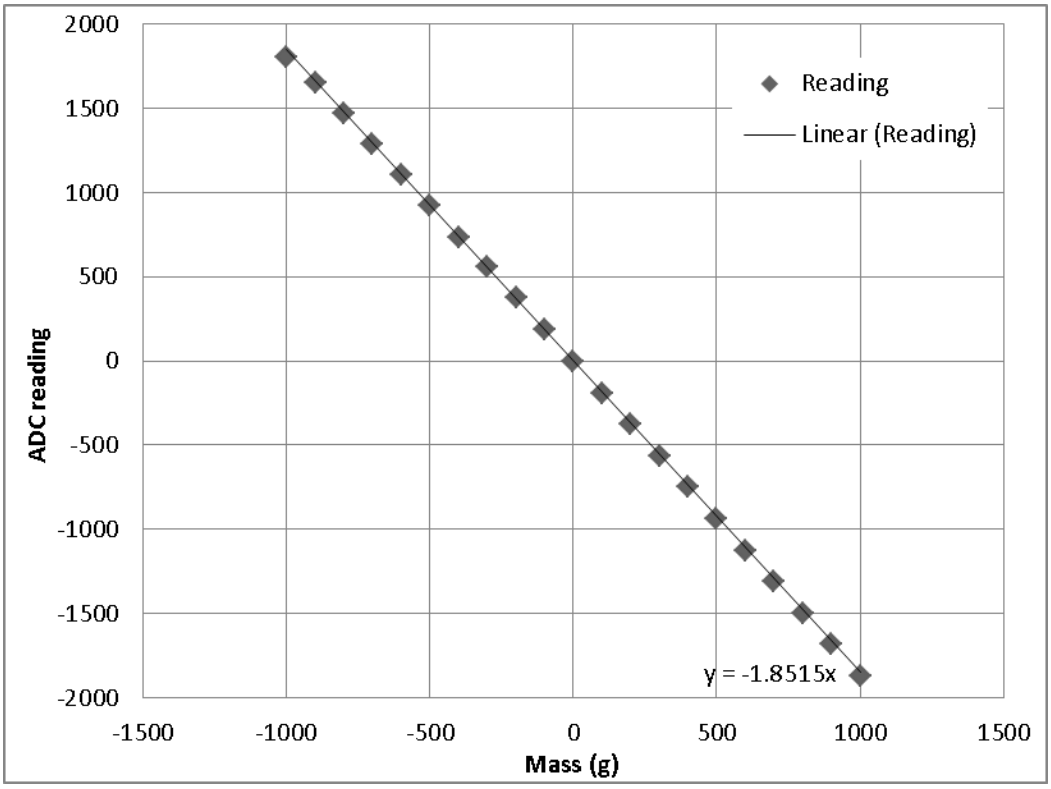

Figure 4.7 Force sensor calibration data points, fitted line and equation

4.1.2 Software Description

The software used in this study, Matlab/Simulink, Quanser QUARC, and Dysim, is described below.

4.1.2.1 Matlab Simulink

Matlab is a computer program developed by Mathworks. It is widely used in universities and industry to carry out calculations and simulations in various situations. One of its main abilities is to solve differential equations. Another advantage is that it provides the Simulink graphical programing and simulation platform. The simulations and experiments in this thesis are all carried out on this platform, in either normal mode or external mode. The version of Matlab used is 2009b.

4.1.2.2 Quanser Realtime Control

Quanser's real-time controller QUARC provides a soft realtime environment in which the experiments are carried out. QUARC 2.0 is used in all the experiments.

QUARC is Quanser's new, state-of-the-art rapid prototyping and production system for real-time control. It is able to generate real-time code directly from Simulink designed controllers for the same PC. QUARC integrates seamlessly with Simulink to allow Simulink models to be run in real time on a Windows target. Its user interfaces enable parameters of the running model to be tuned by changing block parameters in the Simulink diagram and viewing the status of a signal in the model through a Simulink Scope while the model runs on the target. Furthermore, data can be streamed to the Matlab workspace or to a file on disk for off-line analysis. The advantages are summarised as follows [109]:

- Online parameter tuning right from the Simulink diagram
- Plotting right from Simulink
- Code generation for multiple targets from a single Simulink diagram
- Incremental compiling
- The ability to dynamically reconfigure a running system from a supervisory Simulink model
- Data archiving

Quanser also provides a Simulink library that works with its QUARC environment, in which there is a set of Simulink blocks that can be used in combination with each other to interface with the Omni robot in various ways. Within the library, there are two types of block that can be used to interface with the Omni robot, namely the HIL (hardware in the loop) Read/Write blocks and the third party phantom block. The HIL Read and HIL Write blocks are raw data output and input. The HIL Read block outputs encoder counts from the motors and potentiometer ADC values from the wrist joints. The HIL Write block inputs PWM indices to the motor drivers. The third party phantom block (Figure 4.8) comes with integrated input and output ports which are already translated into physical units, *i.e.* joint angle in radians, force input in N, and joint torque input in Nm. There are two modes for input, namely joint torque input and Cartesian force input. The joint torque input mode is chosen for the experiments due to the nature of the controller.

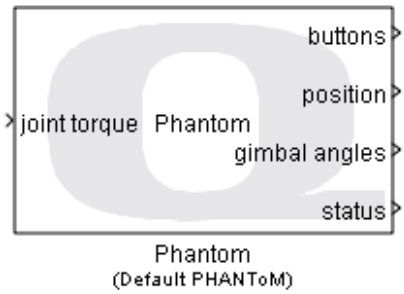

Figure 4.8 Quanser QuaRC Phantom Simulink block

4.1.2.3 Dysim

For the purposes of this study, it is anticipated that one program, developed in-house by Prof M N Sahinkaya, will be used to perform the required system synthesis and modelling tasks. This program, known as Dysim, is a multi-body and multi-physics modelling, simulation and analysis tool. Complicated multi-body systems can be modelled using this tool. The program uses the principles of Lagrangian dynamics [104] to model the complete system; each body in the system is described in terms of the amounts of kinetic and potential energy it possesses accordingly. For the purposes of control scheme design, each mechanical link is considered to be an individual body. Further energy functions can be entered to the system to simulate effects such as energy losses in the system through, for example, friction.

The user must define which of the system variables are dependent and independent once the equations of motion have been derived. Initial conditions for the independent variables must then be specified. These initial conditions can be expressed in terms of basic units (*i.e.* displacement or angle) and first derivatives (*i.e.* velocities). Using this information with the equations of motion, Dysim calculates the corresponding initial conditions for the remaining dependent variables. On some occasions, more than one initial configuration may be possible. For instance, a redundant manipulator can have multiple arm positions for a same end point. To ensure that the desired configuration is identified by Dysim, it may be necessary to state further, approximate, values for the initial conditions of some or all of the dependent variables, to direct the system effectively towards the desired arrangement.

With the construction of the model now complete, the model can be exported from Dysim in the form of an S-function block for use within a Simulink

environment allowing complex analysis of the system to be carried out. Two forms of Simulink block can be created using Dysim.

The first mode is forward dynamics. Through this method, the user must specify the input signal in selected generalised coordinates to the system. The response of the system in generalised coordinates to the input signal is then calculated.

The second mode is the inverse dynamics modelling as shown in Figure 4.9. The user needs to specify the desired output of the system in terms of acceleration of selected generalised coordinates. The input signal in selected generalised coordinates required to generate this output motion is then calculated. The control input and motion-specifying coordinates can be different. However, there must be the same number of motion-specifying coordinates as required control inputs, and this number must be less than or equal to the DOF of the system. In addition to the desired output, the user can also specify other external inputs as in the forward dynamics modelling. It then becomes a hybrid forward/inverse dynamics system, which means the calculated inputs of selected generalised coordinates become the signals that merge with the external generalised inputs to achieve the desired outputs [104].

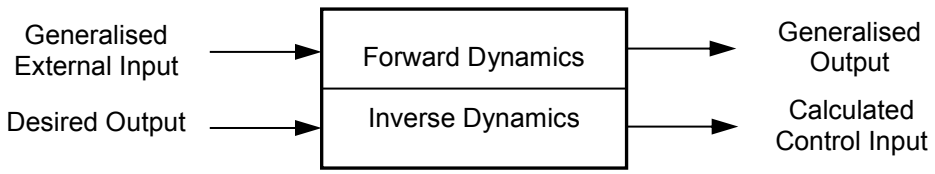

Figure 4.9 Hybrid forward/inverse dynamics block

The inverse dynamics mode also incorporates the new method developed in Chapter 3. The user can also include desired force in addition to the motion input. The constraint force can be set up in the program. A demonstration of the Dysim program can be found in Appendix A.

4.2 Experimental Set-up

In order to test the algorithms and controller developed in this thesis, a set of incrementally more complex experiments are designed:

- **Case I:** one robot manipulate the load in a constrained motion;
- **Case II-a:** remove the constraint and introduce a second robot to hold the position as if there is still a constraint;
- **Case II-b:** both robots move the load together following motion and internal force trajectories.

The details of these different configurations are described in the experiment chapters - Chapter 5 and 6.

The generic experimental test-bed uses two Omni robots which will be configured differently according to the experiment. The configuration consists of two Omni robots. They are considered identical, therefore their physical parameters are assumed to be the same. Figure 4.10 shows the notations of the components; Figure 4.11 shows the notations of the length and distance. Figure 4.12 shows the global coordinates, angle notations, and positive directions.

The length of the load between grabbing points and the distance between robot bases are needed (as shown in Figure 4.11). They can be either measured directly or determined from kinematic relations, known link length, and joint angles. Here both methods are employed to acquire the length and the distance. Hence, an experiment is necessary to estimate the length and the distance kinematically. The links are moved manually in arbitrary motions without any motor commands and load angle and joint angles are recorded. By solving the forward kinematic equations of the rig, the distance and the length are obtained. The values are then used in the parameter estimation process and the controllers.

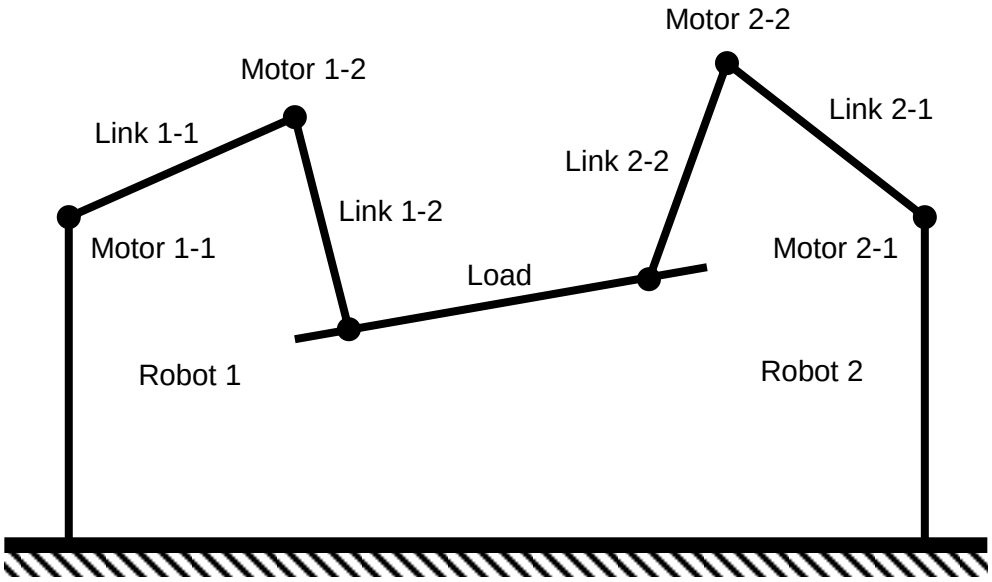

Figure 4.10 Schematic diagram of the experimental set-up with Robot and link definitions

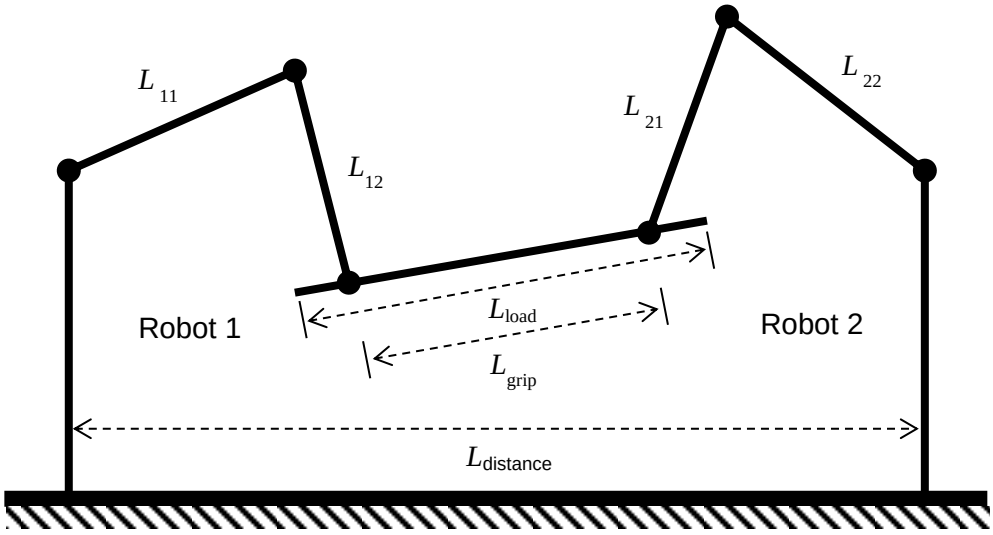

Figure 4.11 Schematic diagram of the experimental set-up with lengths and distances

Table 4.2 Physical parameters of the experimental set-up

Name	L_{11}	L_{12}	L_{21}	L_{22}	L_{load}	L_{grip}	$L_{distance}$
Value (m)	0.132	0.132	0.132	0.132	0.274	0.206	0.475

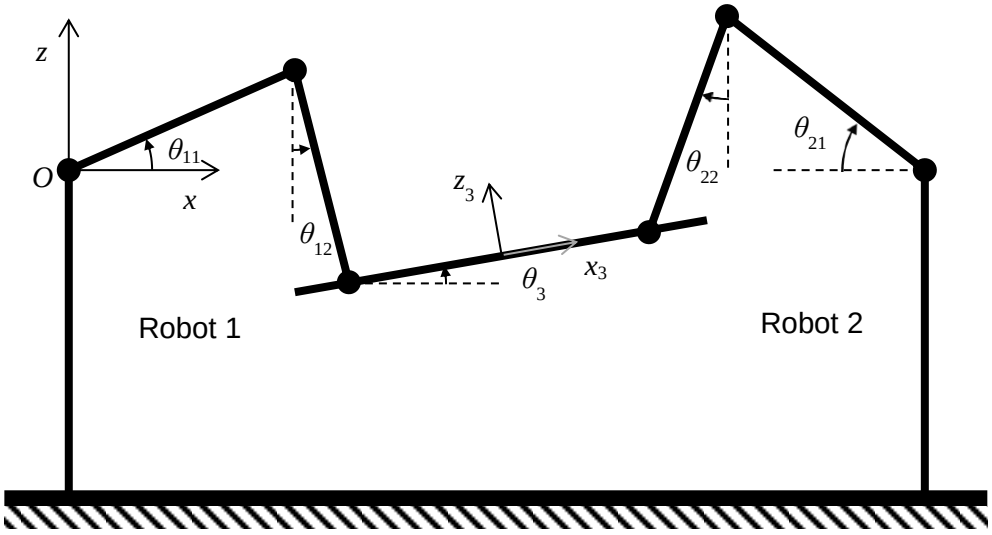

Figure 4.12 Schematic diagram of the experimental set-up with the coordinate frames and generalised coordinates

4.3 Modelling and Parameter Estimation of the Omni Robot Dynamics

This section deals with the dynamic modelling and parameter estimation of the Omni robot. A new approach is presented for designing optimal robot excitation trajectories. It is based on the Schroeder Phased Harmonic Sequence (SPHS), in order to overcome the drawbacks mentioned in Section 2.3. The SPHS signal has fewer parameters to optimise compared to the Fourier series. Then the trajectory is implemented on the Omni robot for parameter estimation. The estimated model is verified

This section is divided into five subsections. Section 4.3.1 describes the generation of the robot dynamics model, and the formulation of Linear Least Squares (LLS) regressor to be applied to identify the robot parameters. Section 4.3.2 introduces the SPHS signal. Section 4.3.3 presents the new approach toward the design of optimal robot excitation trajectories. Section 4.3.4 presents the application of the presented techniques for the experimental estimation of the first three axes of the Omni robot. Section 4.3.5 validates the estimated model against test motions.

4.3.1 Modelling of the Omni Robot Dynamics

As described in Section 3.1.1, the dynamic model of an Omni robot can be stated as:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i, \qquad i = 1, 2, 3 \tag{4.2}$$

and in a general form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{Q} \tag{4.3}$$

These equations are linear in terms of the parameters if they are combined in the so-called Barycentric parameters [101]. Barycentric parameters can be grouped and written as a minimal set of linear equations

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\Theta} = \boldsymbol{\tau} \tag{4.4}$$

where $\mathbf{q}$ is the $n \times 1$ vector of the joint angles, $\boldsymbol{\tau}$ is the $n \times 1$ vector of actuator torques, $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ is the $3d \times r$ regression matrix, depending on the joint angles, velocities, and accelerations, d is the number of data points, $\boldsymbol{\Theta}$ is the $r \times 1$ vector containing the unknown parameters and friction coefficients, r is the number of independent robot parameter sets.

Barycentric model parameterisation is generally known to be difficult to derive [112]. It is done by manually re-grouping the expressions (4.3) according to both common factors and common variables. The resultant regressor matrix $\mathbf{Y}$ and parameters $\boldsymbol{\Theta}$ are listed below.

Joint frictions are also taken into account. The i -th joint frictions are modelled as

$$\tau_{\text{friction},i} = c_{vi}\dot{q}_i + c_{ci}sign(\dot{q}_i) \tag{4.5}$$

where c_v is the viscous friction coefficient, c_c is the column friction coefficient [8].

Add (4.5) to (4.4), the regressor matrix $\mathbf{Y}$ of the first three joints (actuated by motors) of the Omni robot is $\mathbf{Y}^{\mathrm{T}} = [\widehat{\mathbf{Y}}_1^{\mathrm{T}}, \widehat{\mathbf{Y}}_2^{\mathrm{T}}, \widehat{\mathbf{Y}}_3^{\mathrm{T}}]$, where

$$\begin{aligned}\widehat{\mathbf{Y}}_1^{\mathrm{T}} &= [\mathbf{Y}_{1,1}^{\mathrm{T}}, \mathbf{Y}_{1,2}^{\mathrm{T}}, \cdots, \mathbf{Y}_{1,d}^{\mathrm{T}}] \\ \widehat{\mathbf{Y}}_2^{\mathrm{T}} &= [\mathbf{Y}_{2,1}^{\mathrm{T}}, \mathbf{Y}_{2,2}^{\mathrm{T}}, \cdots, \mathbf{Y}_{2,d}^{\mathrm{T}}] \\ \widehat{\mathbf{Y}}_3^{\mathrm{T}} &= [\mathbf{Y}_{3,1}^{\mathrm{T}}, \mathbf{Y}_{3,2}^{\mathrm{T}}, \cdots, \mathbf{Y}_{3,d}^{\mathrm{T}}]\end{aligned} \tag{4.6}$$

where d is the number of data points, and

$$\mathbf{Y}_1^{\mathrm{T}} = \begin{bmatrix} \ddot{q}_1 c(q_3) - \ddot{q}_1 c(2q_2 + q_3) + 2\dot{q}_1\dot{q}_2 s(2q_2 + q_3) + \\ \dots \dot{q}_1\dot{q}_3 s(2q_2 + q_3) - \dot{q}_1\dot{q}_3 s(q_3) \\ \ddot{q}_1 \\ \frac{1}{2}\ddot{q}_1 c(2q_2) - \dot{q}_1\dot{q}_2 s(2q_2) \\ -\frac{1}{2}\ddot{q}_1 c(2q_2 + 2q_3) + \dot{q}_1\dot{q}_2 s(2q_2 + 2q_3) + \dot{q}_1\dot{q}_3 s(2q_2 + 2q_3) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dot{q}_1 \\ sign(\dot{q}_1) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.7}$$

$$\mathbf{Y}_2^{\mathrm{T}} = \begin{bmatrix} 2\ddot{q}_2 c(q_3) + \ddot{q}_3 c(q_3) - \dot{q}_1^2 s(2q_2 + q_3) - \dot{q}_3^2 s(q_3) - 2\dot{q}_2\dot{q}_3 s(q_3) \\ 0 \\ \frac{1}{2}\dot{q}_1^2 s(2q_2) \\ -\frac{1}{2}\dot{q}_1^2 s(2q_2 + 2q_3) \\ \ddot{q}_2 \\ \ddot{q}_3 \\ -s(q_2) \\ -s(q_2 + q_3) \\ q_2 \\ 0 \\ 0 \\ \dot{q}_2 \\ sign(\dot{q}_2) \\ 0 \\ 0 \end{bmatrix} \tag{4.8}$$

$$\mathbf{Y}_3^{\mathrm{T}} = \begin{bmatrix} \ddot{q}_2 c(q_3) - \dot{q}_2^2 s(q_3) - \frac{1}{2}\dot{q}_1^2 s(2q_2 + q_3) + \frac{1}{2}\dot{q}_1^2 s(q_3) \\ 0 \\ 0 \\ -\frac{1}{2}\dot{q}_1^2 s(2q_2 + 2q_3) \\ 0 \\ \ddot{q}_2 + \ddot{q}_3 \\ 0 \\ -s(q_2 + q_3) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \dot{q}_3 \\ sign(\dot{q}_3) \end{bmatrix} \tag{4.9}$$

The ‘s’ and ‘c’ are short for ‘sin’ and ‘cos’ respectively.

The dynamic equation of the Omni robot results in 15 parameters:

$$\mathbf{\Theta} = \begin{bmatrix} M_3 L_2 P_3 \\ I_{1z} + \frac{1}{2}I_{2y} + \frac{1}{2}I_{2z} + \frac{1}{2}I_{3y} + \frac{1}{2}I_{3z} + \frac{1}{2}M_3 L_2^2 + \frac{1}{2}M_2 P_2^2 + \frac{1}{2}M_3 P_3^2 \\ I_{2z} - I_{2y} - M_3 L_2^2 - M_2 P_2^2 \\ I_{3y} - I_{3z} + M_3 P_3^2 \\ I_{2x} + I_{2x} + M_3 L_2^2 + M_2 P_2^2 + M_3 P_3^2 \\ I_{3x} + M_3 P_3^2 \\ g(M_2 P_2 + M_3 L_2) \\ g M_3 P_3 \\ k_{spring} \\ c_{v1} \\ c_{c1} \\ c_{v2} \\ c_{c2} \\ c_{v3} \\ c_{c3} \end{bmatrix} \tag{4.10}$$

where M_i, L_i, P_i, $I_{ix,iy,iz}$, g, i = 1, 2, 3 are link mass, link length, link centre of gravity (COG) distance to previous axis, link inertia moments and gravitational acceleration respectively.

4.3.2 Schroeder Phased Harmonic Sequence (SPHS) Signal

A Schroeder Phased Harmonic Sequence (SPHS) signal is used to generate multi-frequency trajectories. This is a periodic low-peak-factor signal with a given power spectrum [106, 113]. The SPHS signal consists of sine waves with selected frequencies in the following form:

$$s(t) = A_a \sum_{i=1}^{N_H} A_i sin(i\omega_0 t + \varphi_i) \tag{4.11}$$

where ω_0 is the fundamental frequency, N_H is the number of frequency components, A_a is a gain of the overall signal, A_i and φ_i are the amplitude and phase of the i-th harmonic frequency component, which are calculated from the given power spectrum as follows:

$$A_i = \sqrt{\frac{P_i}{2}} \tag{4.12}$$

where P_i is the ratio of the power at $\omega = i\omega_0$ to the total power and can be written as

$$P_i = \frac{A_i^2}{\sum_{j=1}^{N_H} A_j^2} \tag{4.13}$$

and hence

$$\sum_{i=1}^{N_H} P_i = 1 \tag{4.14}$$

The phase of each frequency component is provided by

$$\varphi_i = \varphi_{i-1} - 2\pi \sum_{j=1}^{i-1} P_j \, , i = 2, \dots, N_H \tag{4.15}$$

In the experiments the amplitudes are fixed at A_i = 1 for i = 1, ..., N_H, *i.e.* flat spectrum. Then the fundamental frequency, number of harmonics and overall gain A_a are chosen as the parameters for trajectory optimisation. Figure 4.13 illustrates some examples of SPHS signals with a different number of harmonics and overall gains.

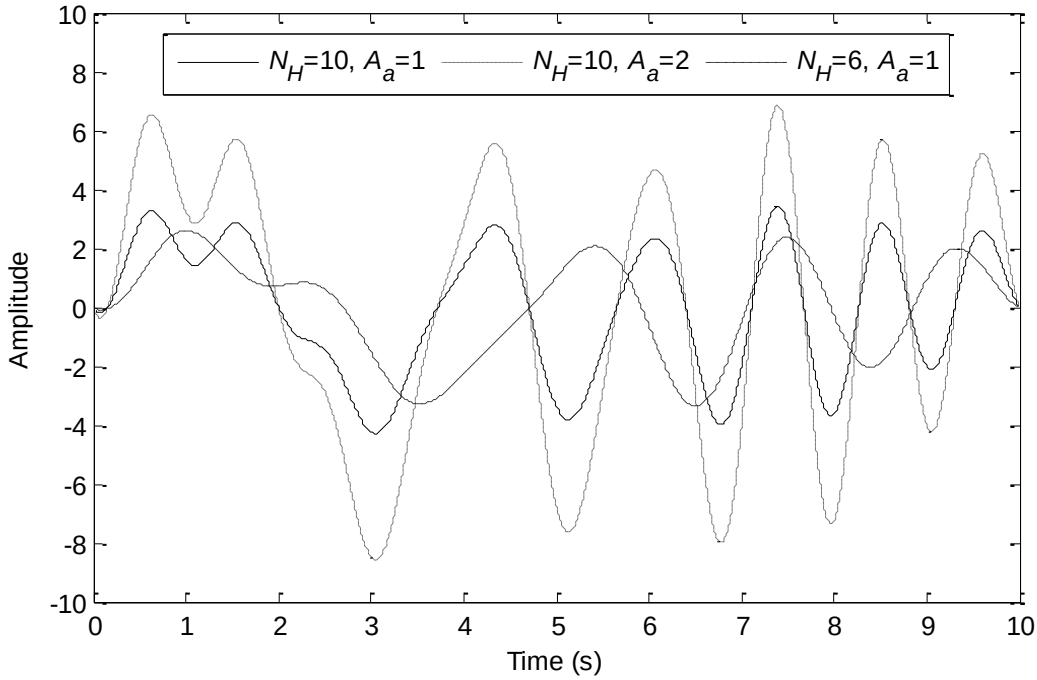

Figure 4.13 Example SPHS signals with different parameters.

It is worth mentioning that Schroeder's formula only works well (for obtaining low peak-to-peak) if all harmonics within the specified range are present, or all odd harmonics within the specified range are present; otherwise, the formula does not work well. If all harmonics are present, then the signal looks like a swept-sine, as can be seen in Figure 4.13.

4.3.3 Optimal Trajectory Design for Identification

Excitation trajectories should have large coverage of the robot workspace and comply with limits in both joint and Cartesian space [114].

The advantages of using SPHS are [11, 12]:

- Controllable frequency spectrum
- Periodic
- Low peak factor (low noise to signal ratio)
- Fewer parameters (ω_0, N_H, and A_a for each signal).

To optimise SPHS signals, a trajectory is required for each DOF. In this case, three trajectories are needed for q_1, q_2, and q_3, of the Omni robot. The aim of the optimisation is to achieve minimum correlation among all three trajectories, thus ensuring maximum coverage of the manipulator workspace.

The fundamental frequency determines the overall period of the motion. Therefore, it is a compromise between low frequency coverage and cost of data collection time. The fundamental frequency should not be too small to avoid long data collection time. It should also not be too large to allow the lower frequency range to be excited.

The trajectory design process consists of three steps:

Step 1: This step ensures that each joint trajectory covers the full motion range within the position, velocity and acceleration constraints. For each joint, the fundamental frequency is set to a small value with a single harmonic component, and overall amplitude to one, *i.e.* ω_0=0.1 Hz, N_H = 1, and A_a=1. Then, the number of harmonics is increased until 90% (for safety reasons) of the maximum joint velocity limits are reached. The method is generally to cope with acceleration or higher derivative limits if known. At the same time, the amplitude of overall signal A_a must be adjusted to satisfy the joint position limits, which are also set to 90% of the actual limits. Figure 4.14 shows an example of the trajectories produced after this step with physical and trajectory limits in Table 4.3 and Table 4.4, respectively. The mean value of each signal represents the midpoint of each corresponding joint angle range.

Table 4.3 Actual Ranges of Actuated Joints

Joint No.	Position (degree)	Velocity (degree/s)
1	[33.7 146.2]	310
2	[-15.6 73.8]	300
3	[86.7 119.3]	350

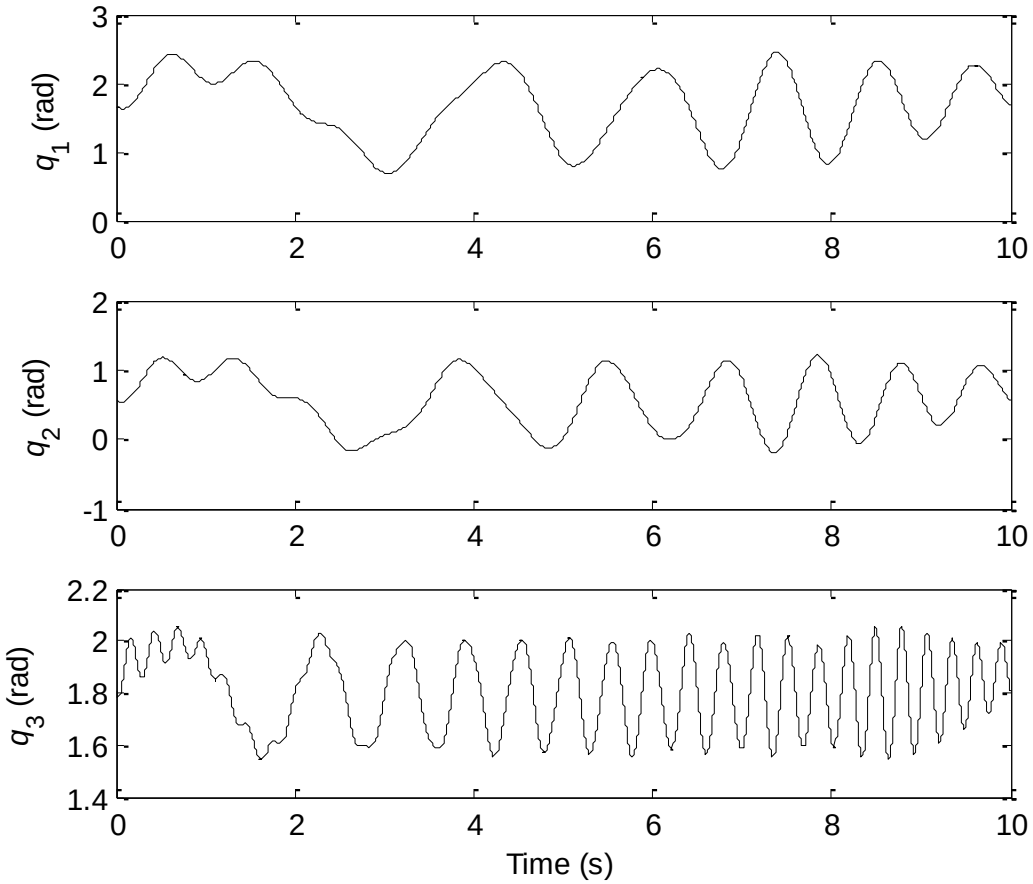

Figure 4.14 One period of SPHS signal of q_1, q_2 and q_3 after step 1.

Table 4.4 Trajectory Parameters and Limits

Joint No.	ω_0 (Hz)	N_H	A_a	Position (degree)	Velocity (degree/s)
1	0.1	10	0.2232	[39.3 140.6]	256.4
2	0.1	12	0.1612	[-11.1 69.3]	258.8
3	0.1	38	0.0347	[88.3 117.7]	314.1

Step 2: Trajectories should have low correlation in order to cover more space in fixed motion ranges. To facilitate this, a trajectory is chosen (e.g. q_1) and the others (q_2 and q_3) are shifted in time (delayed) with respect to the selected trajectory. The optimal delays will be those that reduce the correlation coefficients between q_2 and q_1 ($C_{1,2}$), and q_3 and q_1 ($C_{1,3}$). Arbitrarily small delays are incrementally added, up to one period. As shown in Figure 4.15, several zero correlation points can be found for both correlation coefficients (16 delay values for q_2 and 12 for q_3 in this case). Then these optimal delay values are used to calculate the correlation coefficients between q_2 and q_3 ($C_{2,3}$). Then, a set of delays (five in this example) that result in the smallest $|C_{2,3}|$ values are selected for the final step.

Step 3: The sub-set of signals from step 2 are used to calculate the determinant of the covariance of the regressor matrix, $\mathbf{Y}^T\mathbf{Y}$. The combination with the maximum determinant is chosen as the optimal excitation trajectories, in order to minimize the standard deviation of the estimated coefficients.

The three-stage optimisation results in two delay values for two of the trajectories. In this case, 93.17% and 86.84% are the delays for q_2 and q_3 respectively. The cross plot between the optimised trajectories q_2 and q_3 is shown in Figure 4.25. The figures on the right hand side illustrates that joint spaces are well covered.

Table 4.5 Optimal Trajectory Parameters

Joint No.	ω_0 (Hz)	N_H	A_a	Delay (%)
1	0.1	10	0.2232	0
2	0.1	12	0.1612	93.17
3	0.1	38	0.0347	86.84

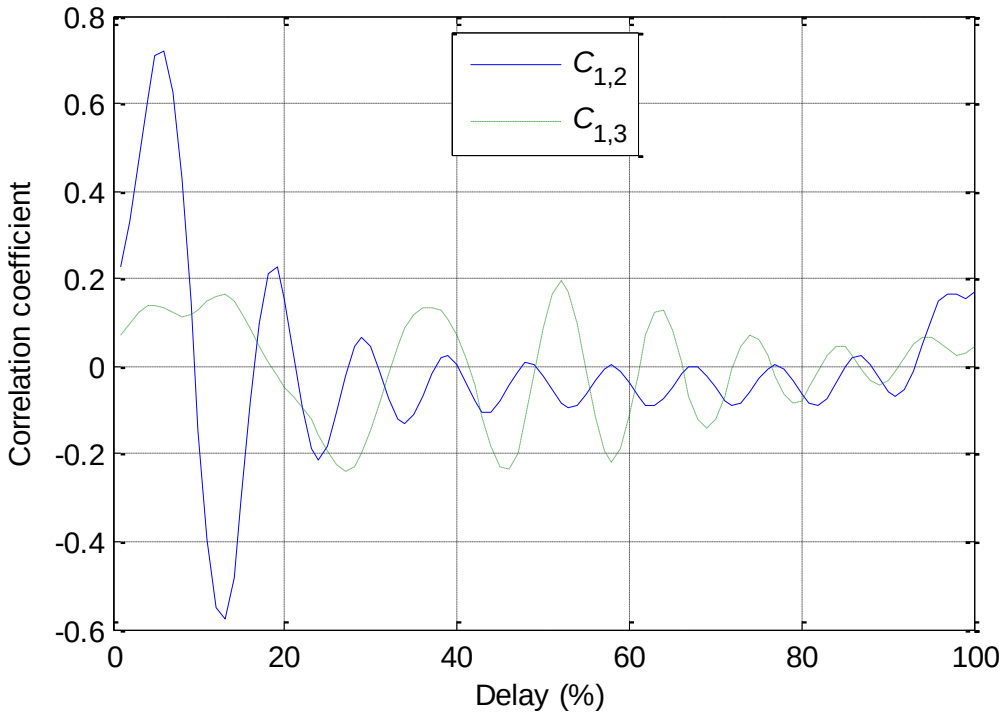

Figure 4.15 Correlation coefficients between q_1 and q_2 ($C_{1,2}$), and q_1 and q_3 ($C_{1,3}$) as a function of delays on q_2 and q_3 respectively.

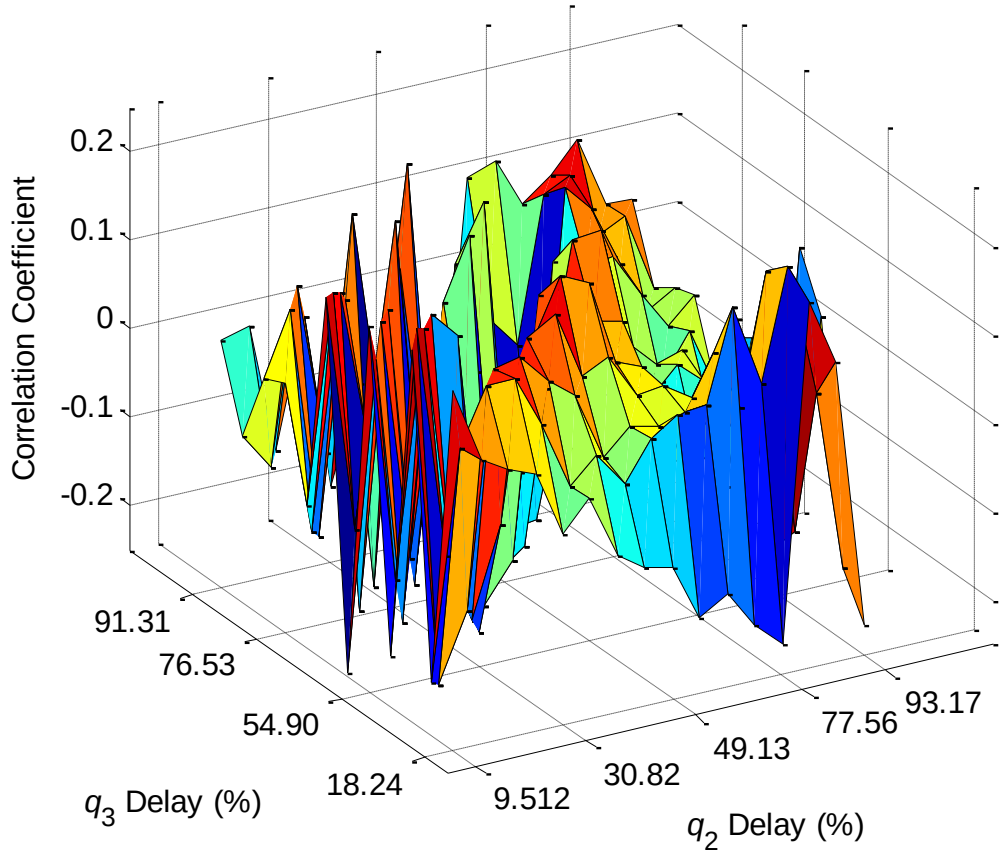

Figure 4.16 A heat map showing correlation coefficients between q_2 and q_3 ($C_{2,3}$) using delay values which have zero correlation with q_1.

It should be noted that the difficulty in performing Step 2 increases when there are more DOF (for instance, six). Take an ordinary industrial manipulator as an example; it consists of six DOF. The first three define the position of the wrist centre relative to the robot base, and the last three define the orientation of the tool around the wrist centre. This method can be applied to the first three and the last three DOF separately. Because those two parts usually possess parameters whose values are in different order of magnitude.

4.3.4 Application to Estimation of the Omni Robot

A few issues that are unique to the Omni robot need to be dealt with before the method mentioned above can be applied, namely angle limits and joint velocity limits.

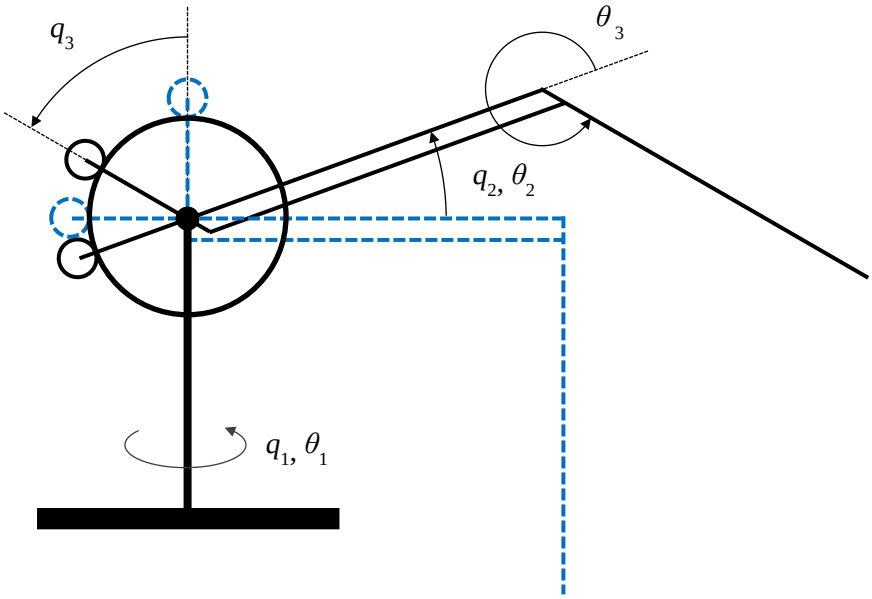

Figure 4.17 Phantom Omni schematic diagram (only the first three links)

Because of the mechanical design of the Omni robot (Figure 4.17), the angle limits between Joint 2 and Joint 3 have a unique shape as shown in Figure 4.18. Unlike typical manipulators, the Link 3 is powered by the motor inside Link 1 through a cable connected parallel mechanism. Therefore, both ranges are dependent upon each other's position. To use the method described in the previous section, the joint space trajectory has to be inside the shape and cover as much of the space as possible. In order to achieve this, one needs to find the angle limits that do not violate physical limits and at the same time cover the widest range in both joints, *i.e.* to find the biggest rectangle inside the shape.

A Monte Carlo simulation has been used to determine the size and position of the rectangle inside this shape.

In the q_2-q_3 plane, a rectangle can be defined by its centre position and size in the unit of angle. By observing the shape, it is found that the biggest rectangle is most likely to appear around the graphical centre of the shape. Therefore, the centre is used as a reference point for the Monte Carlo simulation.

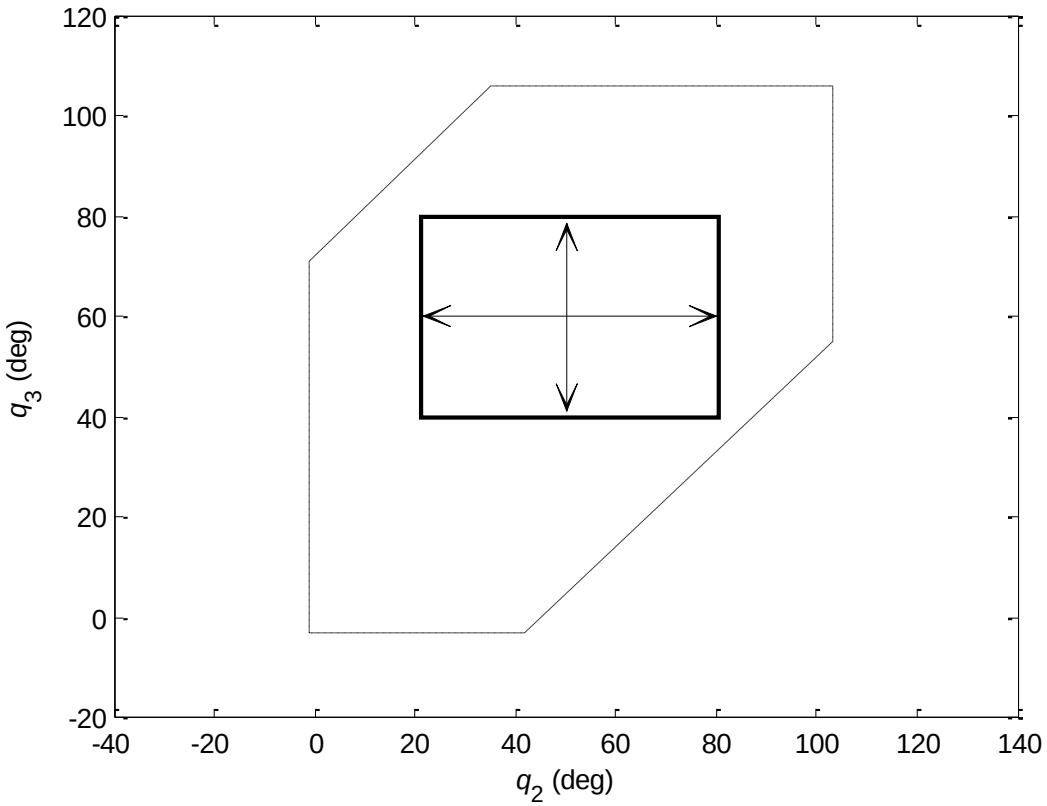

Figure 4.18 Physical limits between Joints 2 and 3 of the Omni robot in absolute angle (dashed line). The centre and size of a rectangle need to be determined for Step 1.

To find the graphical centre of the shape, the Matlab command 'getframe' is used. It captures the image from the plot and generates an image region object. Then a command 'regionprops' from Image Processing Toolbox is used to find the centre of the shape utilising the object property 'Centroid'. The resultant centre coordinates are q_2= 46.2564 deg and q_3=57.0200 deg.

Using this centre as a reference point, the four parameters for the simulation are two shrink ratios (4.16) and two distances of centre shift (in degree). The two shrink ratios act on each of the physical ranges. The centre shifts are the distances that the rectangle centre moves in each dimension. The simulation range of the parameters is obtained by trial and error and is listed in Table 4.6. Any ratio larger than 75% results in no trajectory in range, and any ratio smaller than 55% produce too small rectangles that cannot cover well the range. The correlation coefficients are calculated using Matlab command 'corrcoef'. Figure 4.19 and Figure 4.20 show the simulation results. In both figures, the columns of small figures stand for different shrink ratios on the q_2 range, and the rows stand for different shrink ratios on q_3. In any of the small figures, the axes stand for the distance the rectangle centre moves in each dimension. In Figure 4.19, the

colour stands for the number of trajectories in range when the parameters are changed, the cooler the more, ranging from 0 to 100. In Figure 4.20, the colour stands for the minimum correlation coefficients of the number of trajectories in range at that point.

$$\text{Shrink ratio} = \frac{rectangle\ size}{maximum\ angle\ range} \times 100\% \tag{4.16}$$

Table 4.6 Monte Carlo simulation parameter initial values on rectangle centre and size

Parameter	**Width Ratio**	**Height Ratio**	**Centre Shift Right (deg)**	**Centre Shift Up (deg)**
Range	55% - 75%	55% - 75%	0 - 10	0 - 10

Choosing parameters is a compromise between keeping potential trajectories and covering maximum possible range. In Figure 4.19 and Figure 4.20, a trend can be observed showing that the smaller the rectangle the more trajectories lie within the physical limits, hence smaller correlation coefficients can be found among those trajectories. Although more trajectories and smaller correlation coefficients are found in even smaller rectangles (in the top left corners of both figures), the aim of this whole process is to find the largest rectangle. Hence, more simulations are carried out in the bottom right corner between 70% and 75% shrink ratios. Figure 4.21 and Figure 4.22 show the results.

From Figure 4.21 and Figure 4.22 one can observe that the shrink ratio 71% on both q_2 and q_3 is the largest rectangle which both has more than 10 trajectories and has lower correlation coefficients (shown in blue). Therefore, the size of the rectangle can now be obtained.

However, in terms of the centre shifts, there are a number of identical trajectories, in a diagonal sense as shown in Figure 4.21 and Figure 4.22. This is because of the special shape of the physical limits. Three trajectories are plotted in Figure 4.23, they are from the two ends and the centre on the diagonal line.

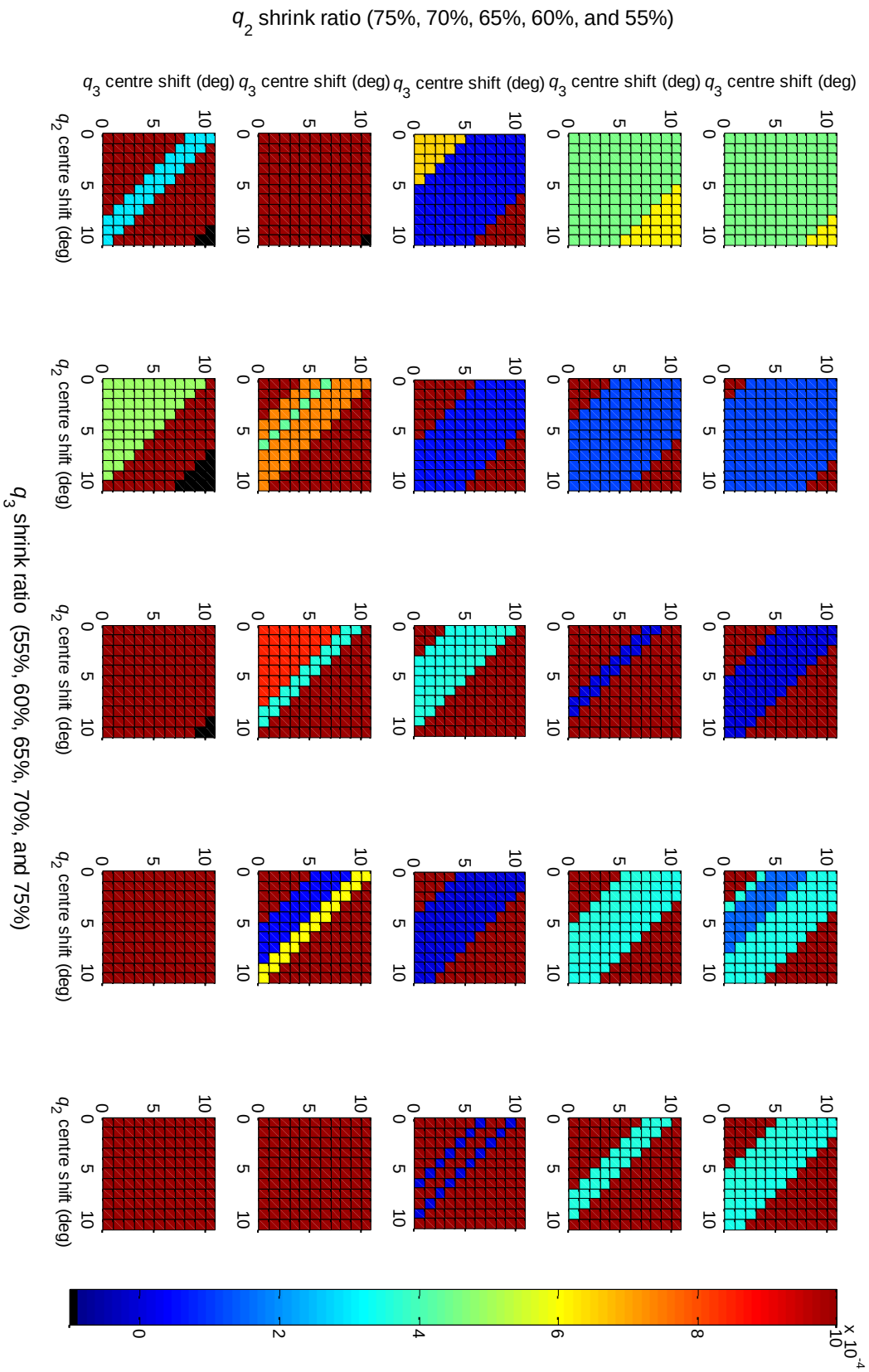

Figure 4.19 Monte Carlo simulation results in finding the joint angle ranges. The colour stands for the number of trajectories that are in the hard limits when the size and centre of the rectangle change.

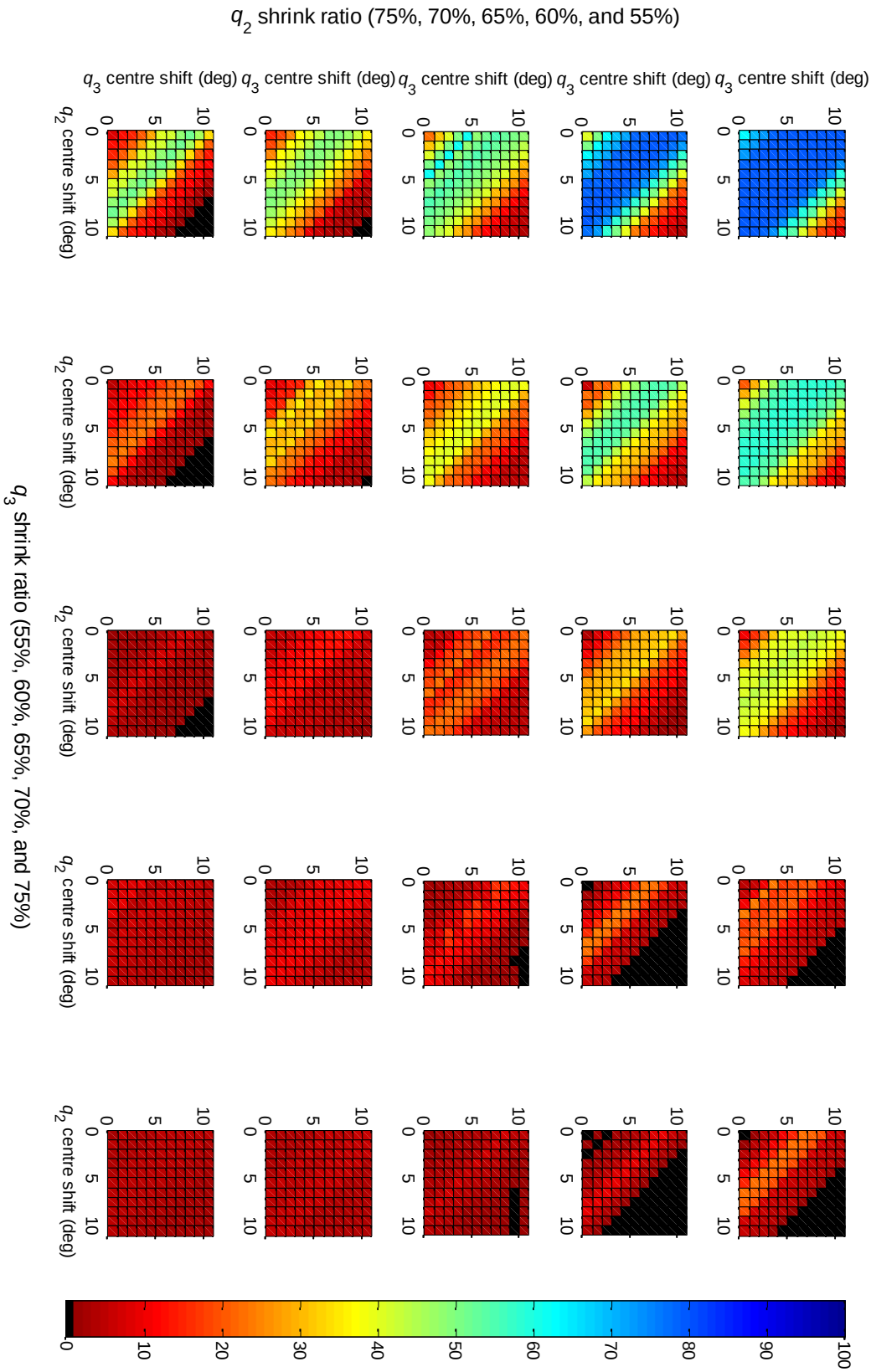

Figure 4.20 Monte Carlo simulation results in finding the joint angle ranges. The colour stands for the minimum correlation coefficient of the trajectories found in Figure 4.19.

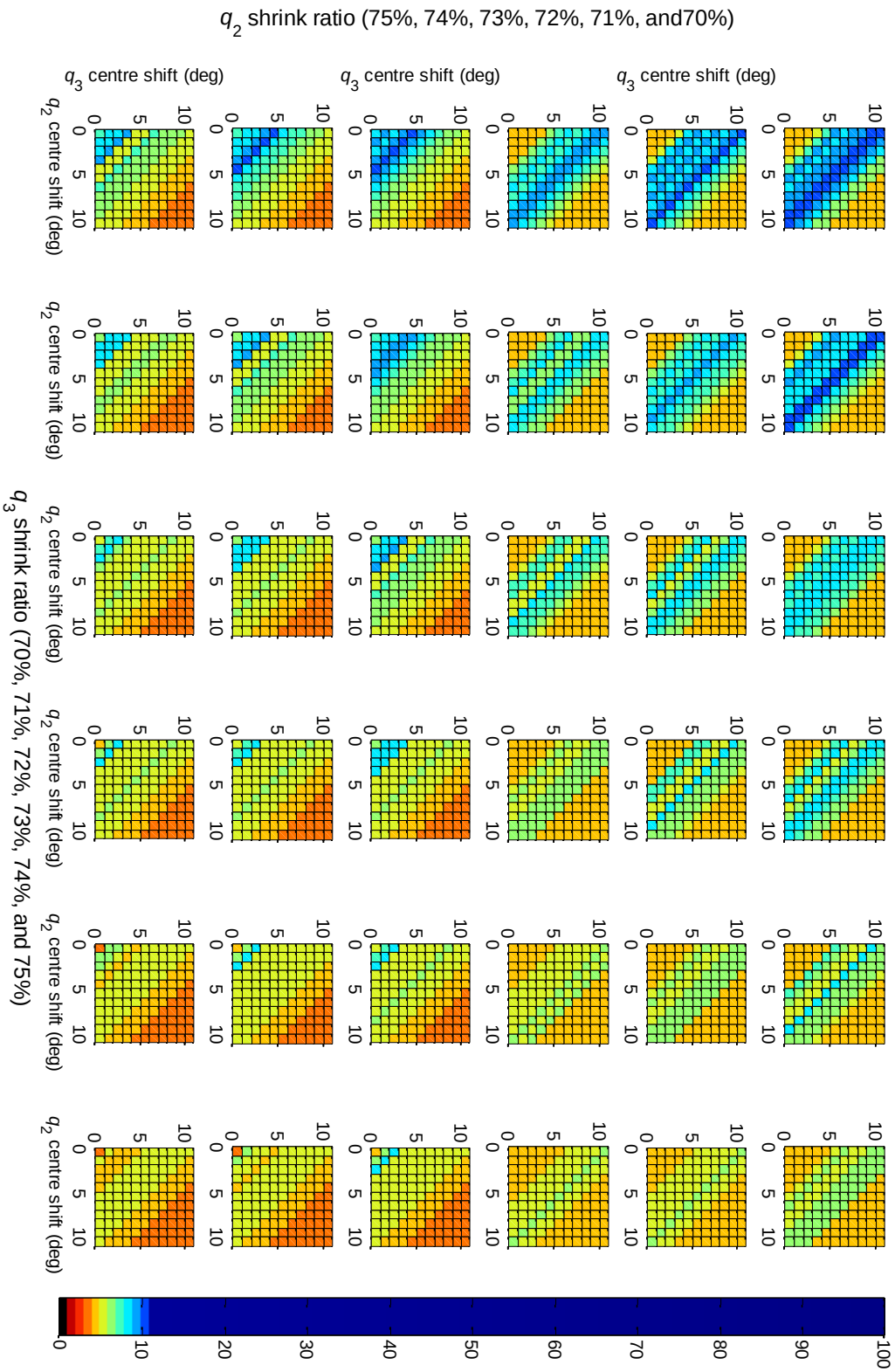

Figure 4.21 More Monte Carlo simulation results that zoom into the bottom right corner of Figure 4.19

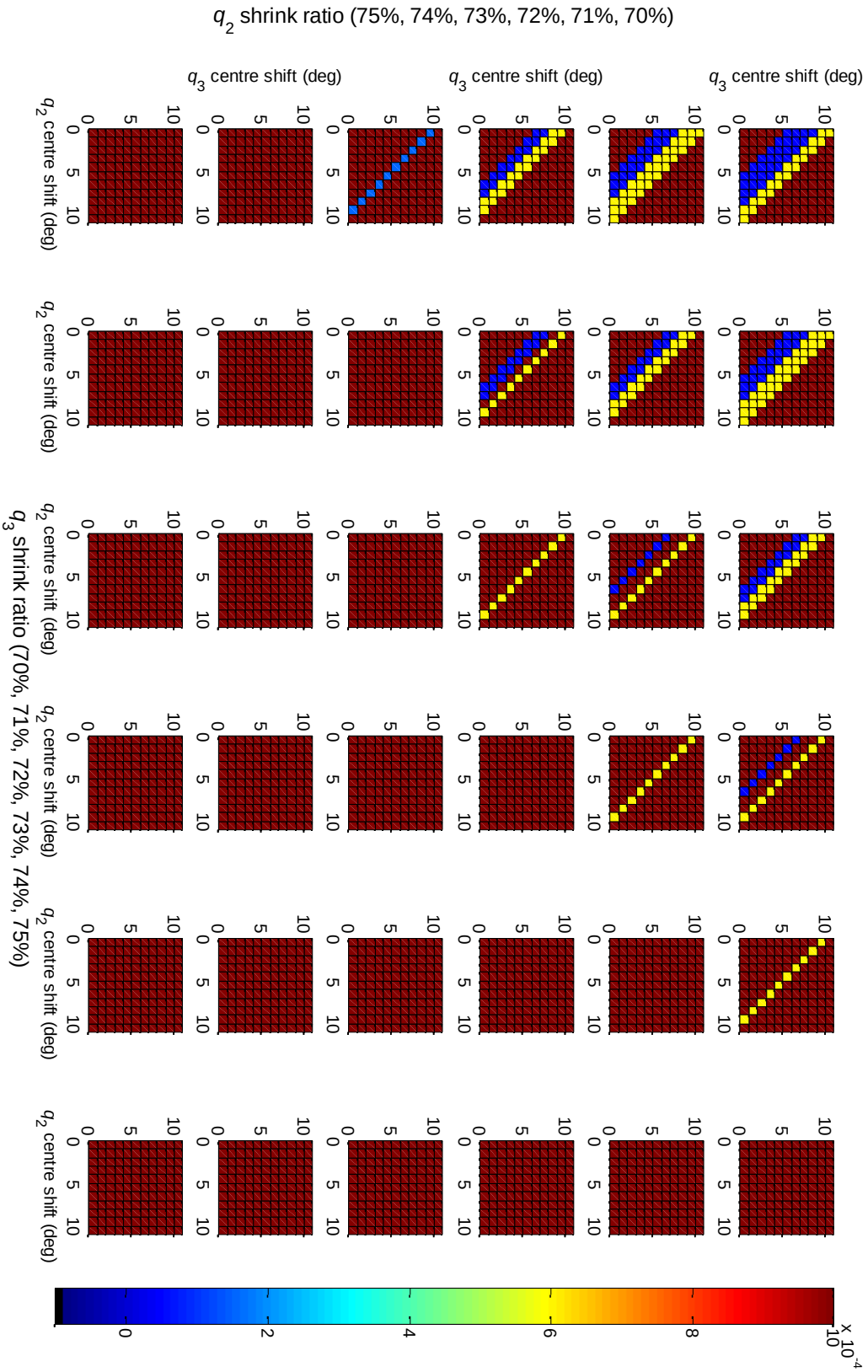

Figure 4.22 More Monte Carlo simulation results that zoom into the bottom right corner of Figure 4.20

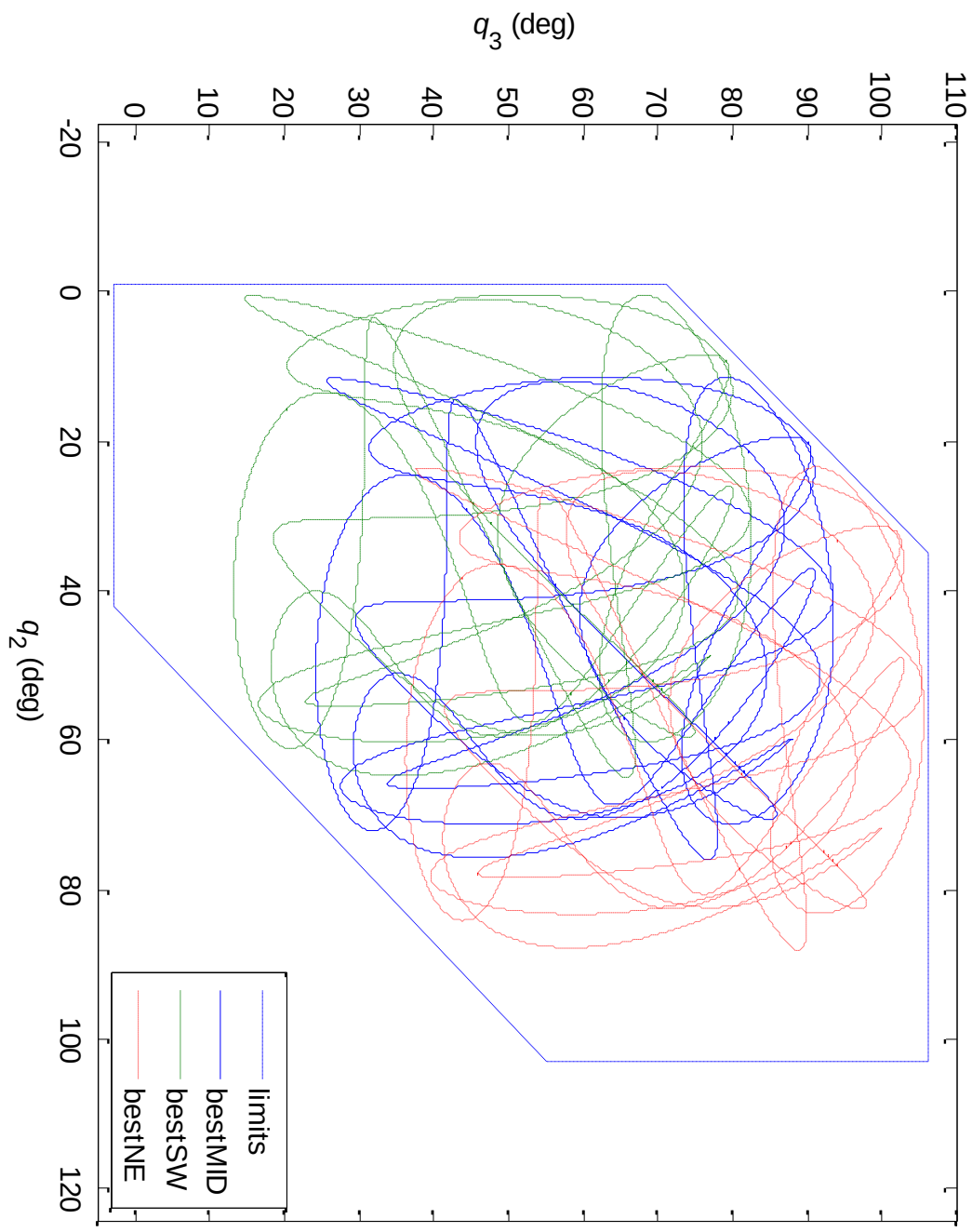

Figure 4.23 Three trajectories with the smallest correlation coefficients

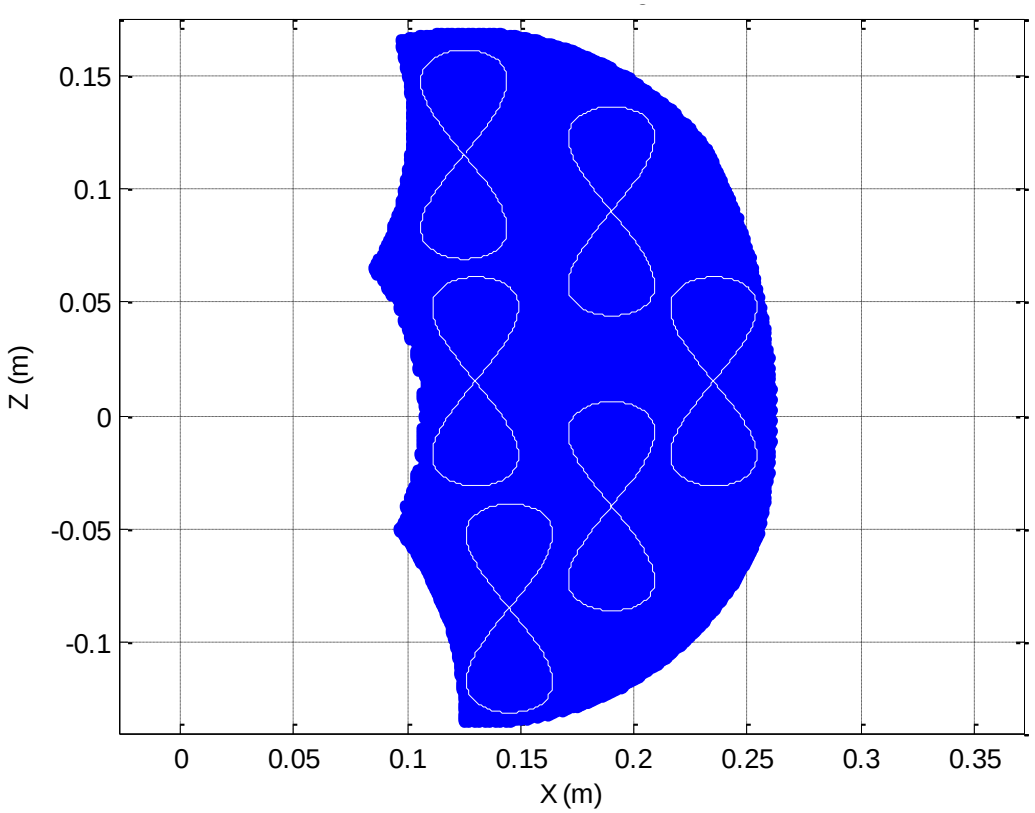

Figure 4.24 Test of motions in different regions in Cartesian space

To choose the best trajectory from these three shown in Figure 4.23, another test is designed. Six small regions across the working range are selected to carry out a same profile figure-of-eight motion, as shown in Figure 4.24. The centre coordinates of the figure-of-eight are listed in Table 4.7. The joint angles and torque demands are recorded. The measurements are fed into different models estimated from different trajectory positions - SW, MID and NE. The calculated joint torques are then compared with the actual torque demands. The MID model shows the best correlations (Table 4.8), hence the MID trajectory is chosen.

Table 4.7 Figure-of-eight centre coordinates

x (m)	0.125	0.190	0.030	0.230	0.190	0.145
z (m)	0.115	0.090	0.015	0.015	-0.040	-0.085

Table 4.8 Correlation coefficients between actual torques and calculated model torques from different trajectory positions

	MID	**SW**	**NE**
Correlation coefficient	0.8568	0.8539	0.8429

The resultant joint angles are shown in Figure 4.25 and Figure 4.26, where the difference between high correlation and low correlation is clearly shown. The non-optimised trajectories have less coverage of the joint space. Figure 4.27 shows the corresponding Cartesian trajectories. The set of non-optimised trajectories is used to estimate the parameters for a model called the non-optimised model.

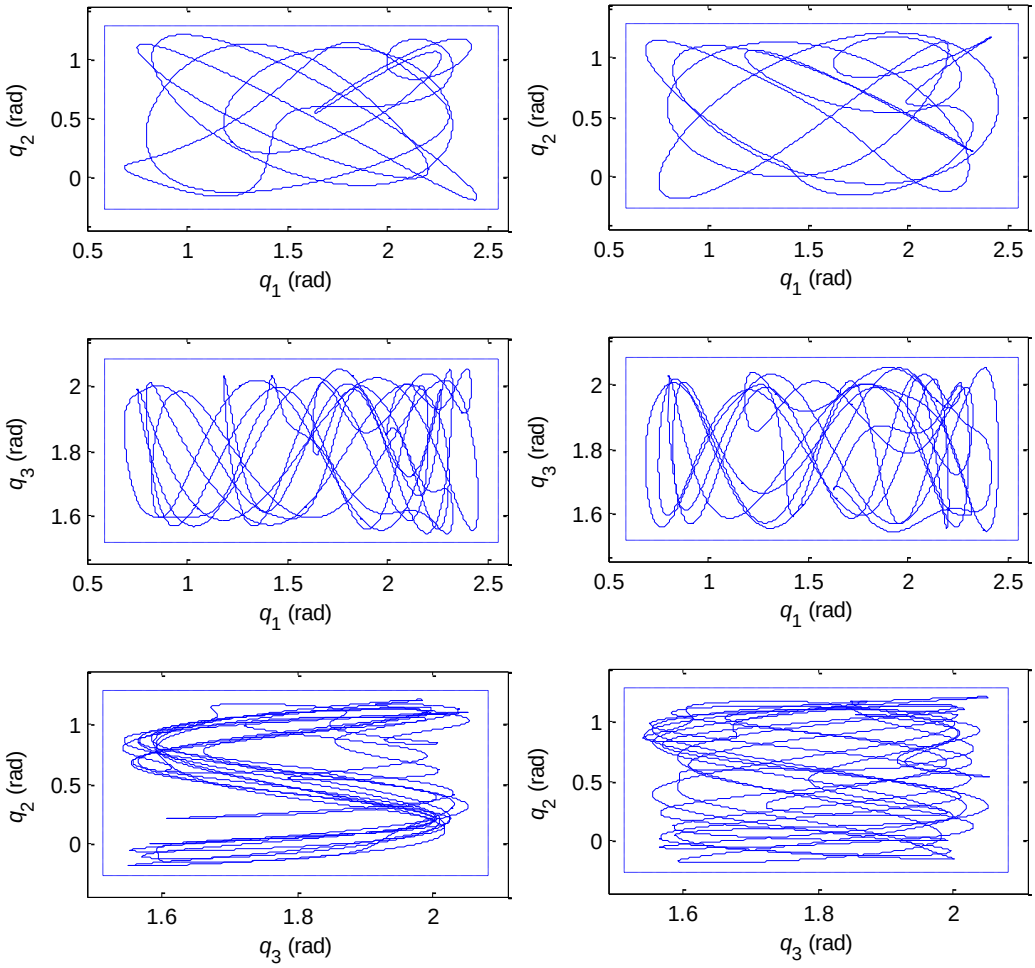

Figure 4.25 Cross plots among three trajectories for q_1, q_2 and q_3 before (left) and after (right) correlation minimization. The dotted boxes are the joint position limits.

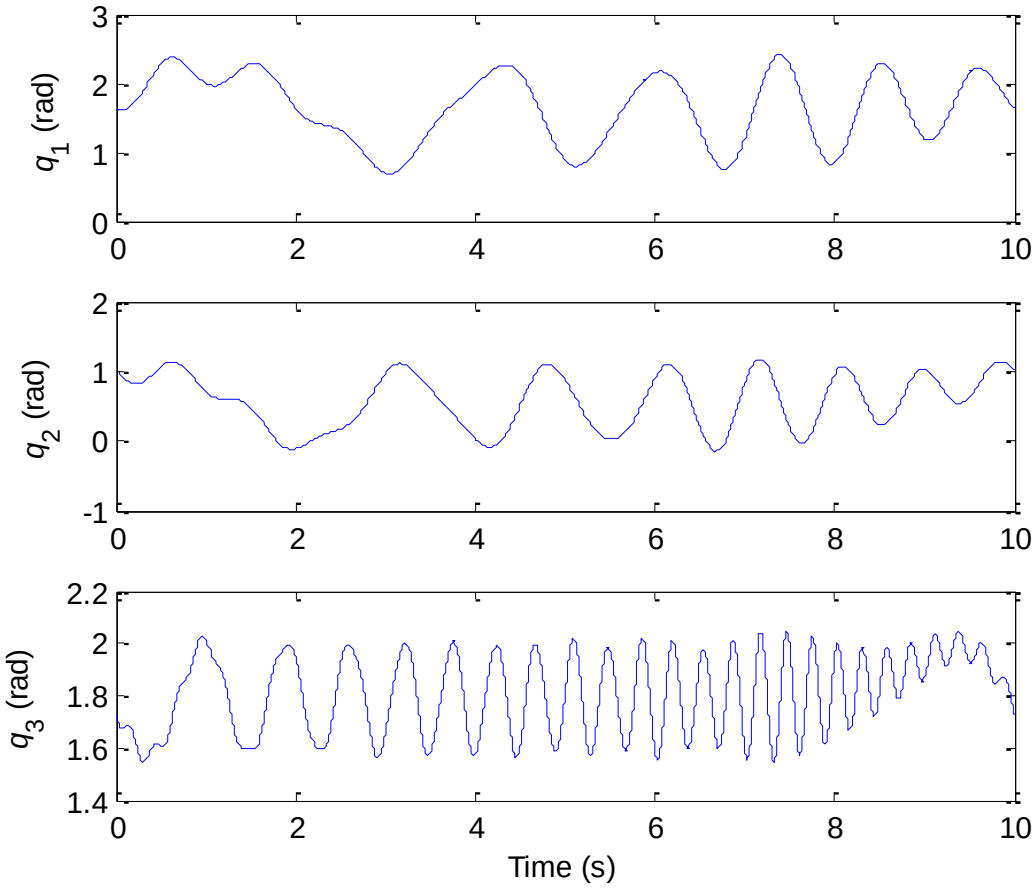

Figure 4.26 Final trajectories after Step 3

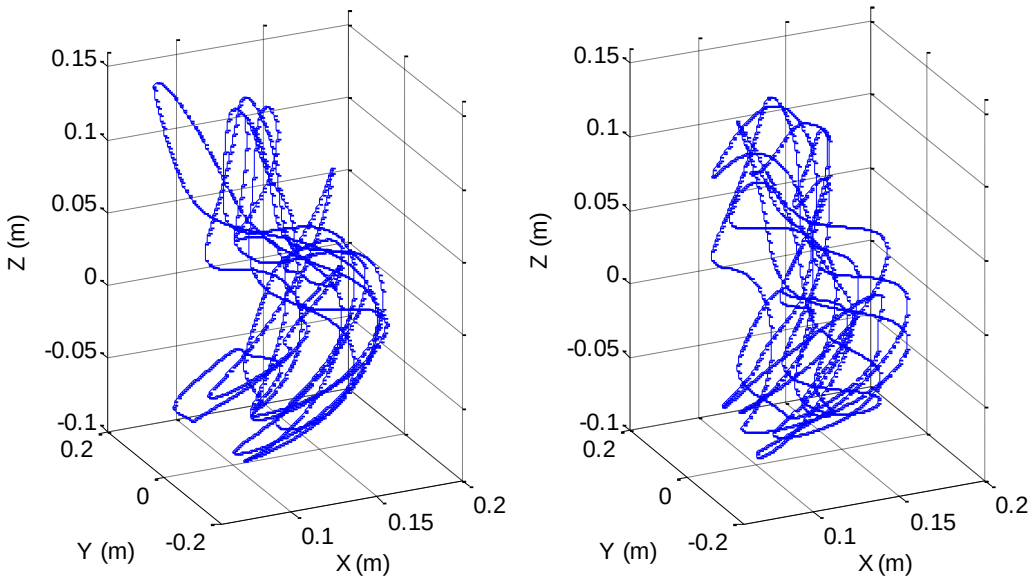

Figure 4.27 Trajectory of the Omni robot's end point in one period in Cartesian space before (left) and after (right) correlation minimization

The robot is working under joint angle feedback control with PID controllers. Demand trajectories are given and the measured joint angles are recorded in a

Matlab/Simulink environment. The measurement is taken for 20 periods. Due to lack of torque measurement, the torque demands are recorded instead and later transformed into joint torques [115]. The data are then averaged. The angles are fed through an integrator based second-order low-pass filter which can also provide velocity and acceleration with relatively low noise (Figure 4.28). The sampling rate is 1000 Hz.

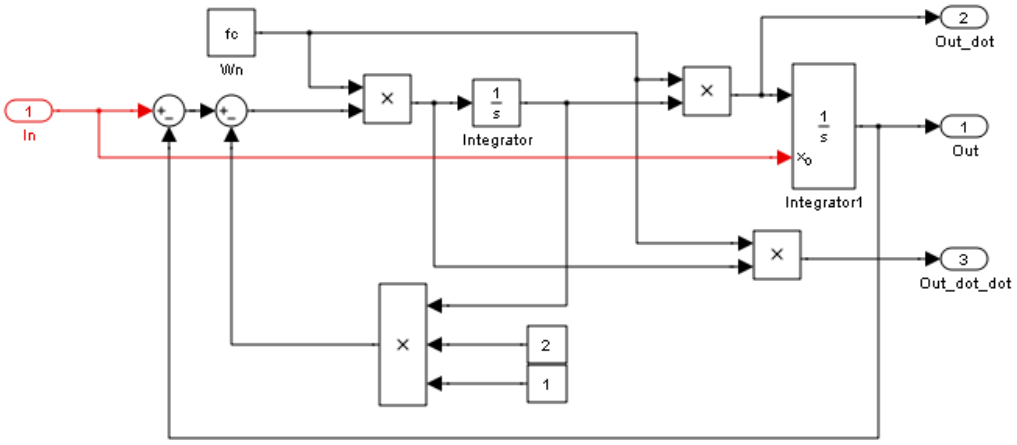

Figure 4.28 An integrator based second-order low-pass filter with derivative outputs

Estimations are carried out using the linear least squares. The estimated parameters with corresponding standard deviation are listed in Table 4.9. For comparison purposes, the model using these parameters is called the optimised model.

Table 4.9 Parameters (4.10) estimated from optimal excitation (with standard deviation in brackets)

$\hat{\theta}_1$	$1.42\ (0.08)\times10^{-3}$	c_{v1}	$-1.86\ (1.88)\times10^{-3}$
$\hat{\theta}_2$	$2.37\ (0.16)\times10^{-3}$	c_{c1}	$3.94\ (0.41)\times10^{-2}$
$\hat{\theta}_3$	$-2.36\ (0.39)\times10^{-3}$	c_{s1}	$3.54\ (0.60)\times10^{-2}$
$\hat{\theta}_4$	$1.50\ (0.32)\times10^{-3}$	c_{v2}	$2.00\ (1.77)\times10^{-3}$
$\hat{\theta}_5$	$4.63\ (0.09)\times10^{-3}$	c_{c2}	$5.04\ (0.39)\times10^{-2}$
$\hat{\theta}_6$	$1.58\ (0.03)\times10^{-3}$	c_{s2}	$5.16\ (0.56)\times10^{-2}$
$\hat{\theta}_7$	$2.92\ (0.51)\times10^{-1}$	c_{v3}	$4.50\ (1.29)\times10^{-3}$
$\hat{\theta}_8$	$9.92\ (0.32)\times10^{-2}$	c_{c3}	$7.83\ (2.97)\times10^{-3}$
$\hat{\theta}_9$	$1.76\ (0.43)\times10^{-1}$	c_{s3}	$6.63\ (5.09)\times10^{-3}$

4.3.5 Estimated Model Validation

A test motion is used to validate the estimated models. The motion is generated from arbitrarily chosen sine waves. For all the joints, the amplitude is 20 degrees. Frequencies are set to 0.4, 0.5, and 1.2 hertz for Joints 1, 2 and 3 respectively. Table 4.10 shows the measured and estimated torques of both optimised and non-optimised models. The RMS of the torque residues is shown in Table 4.10. It shows that the optimised model improves the overall torque predictions, reducing at least by half the RMS for joint 2 and 3.

The estimated models have been tested in a control set-up where the optimised and non-optimised models are used as feedforward compensators as shown in Figure 4.30. The feedback control gains are the same for both controllers. Figure 4.31 illustrates a figure-of-eight demand trajectory and the achieved trajectories for both models. The period of the motion is one second. Table 4.10 shows the position RMS error in both x and y-axes. As can be seen, the optimised model follows the trajectory more accurately, and thus is a preferred model.

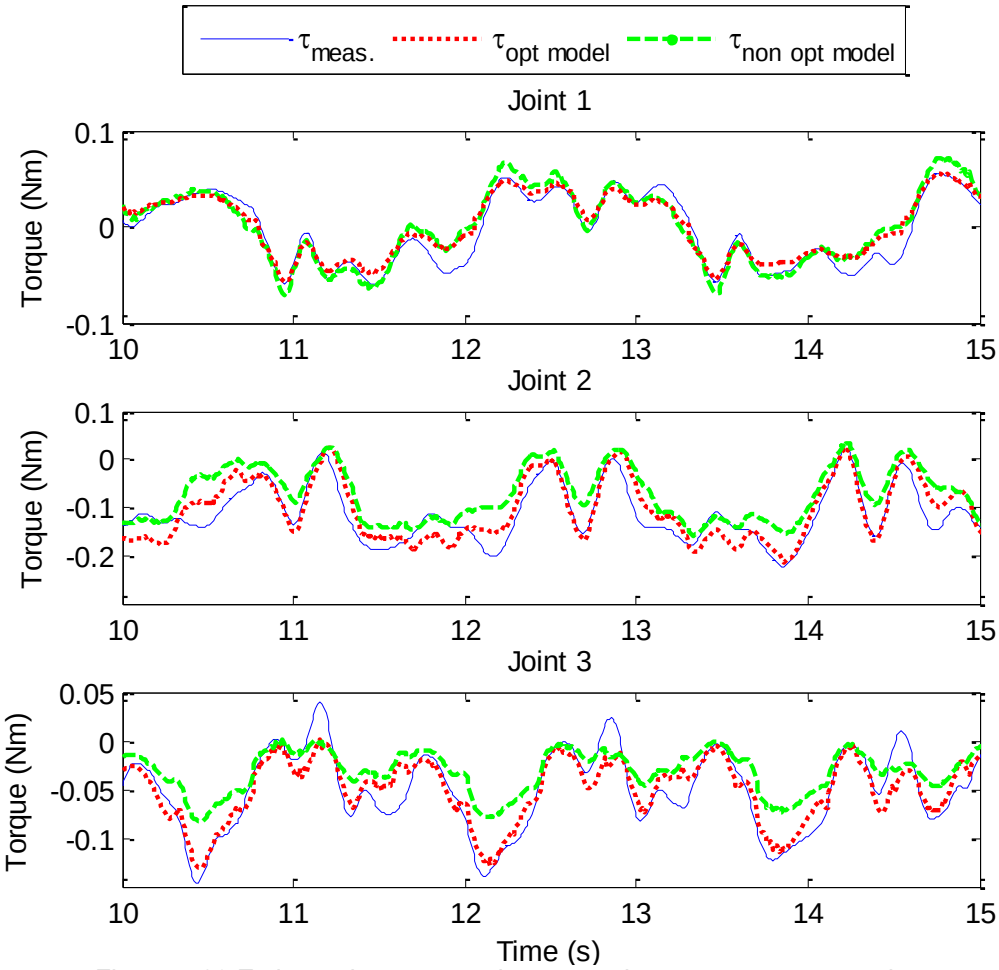

Figure 4.29 Estimated torques and measured torques on a test motion

Table 4.10 Torque Residue RMS (Nm)

Model	Joint 1	Joint 2	Joint 3
Optimised	0.01186	0.02868	0.01751
Non-optimised	0.01326	0.06673	0.03110

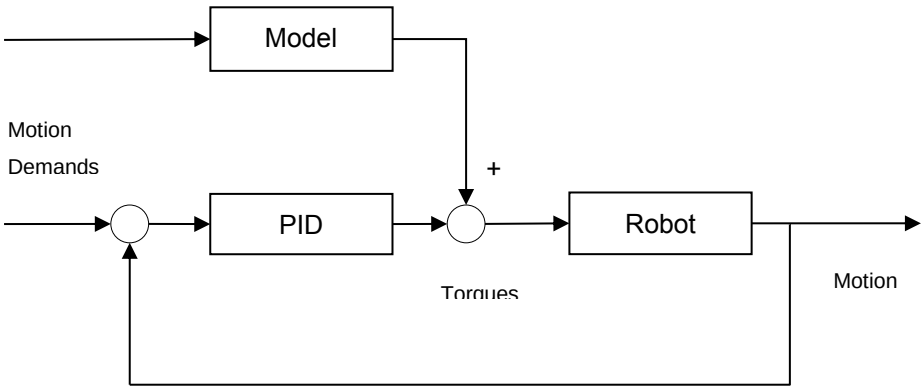

Figure 4.30 Control system diagram with the feed-forward compensator

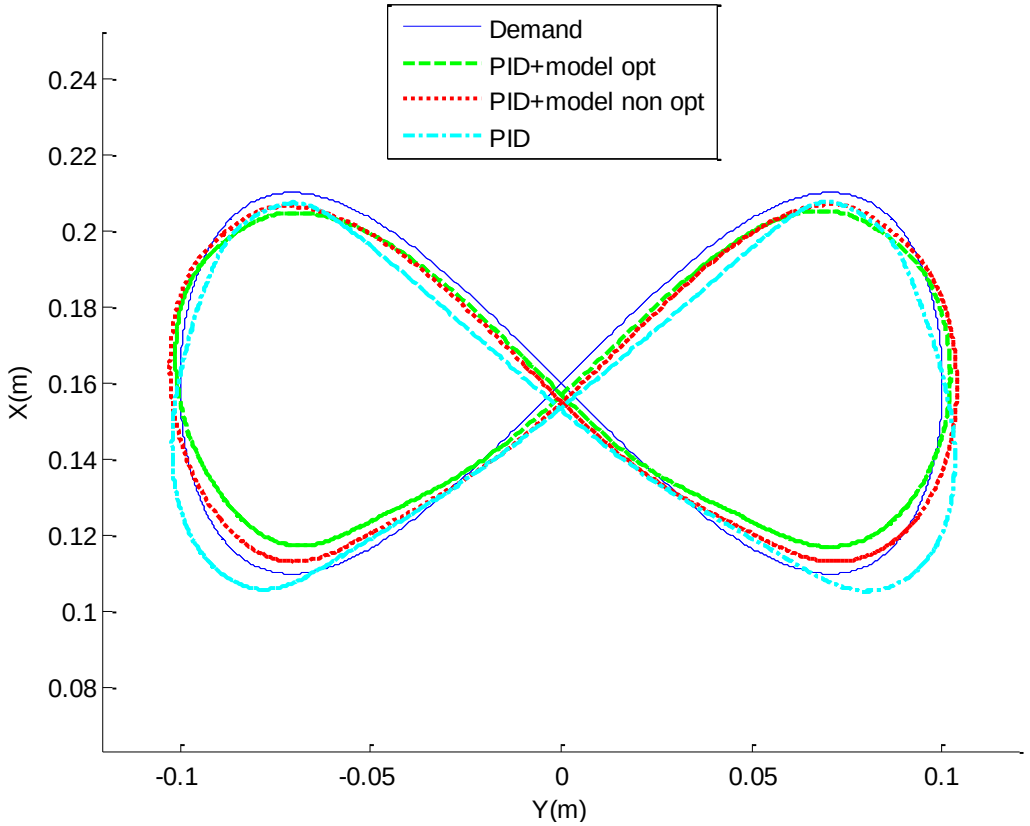

Figure 4.31 Figure-of-eight motion of the wrist centre in a horizontal plane

Table 4.11 Cartesian Position Error RMS (m)

Controller Type	X	Y
PID + optimised model	0.0118	0.0096
PID + non-optimised model	0.0123	0.0101
PID only	0.0136	0.0139

The correlation coefficients between the total torque and individual components are calculated to show the performance of the feedforward model. It can be seen that in general the controller performs better with feedforward compensation. Furthermore, the feedforward component contributes significantly, as shown in Figure 4.32 to Figure 4.35, Table 4.12, and Table 4.13. These figures and tables show clearly that the model provides most of the torque demands, whereas the PID controller provides compensation mainly for friction modelling errors.

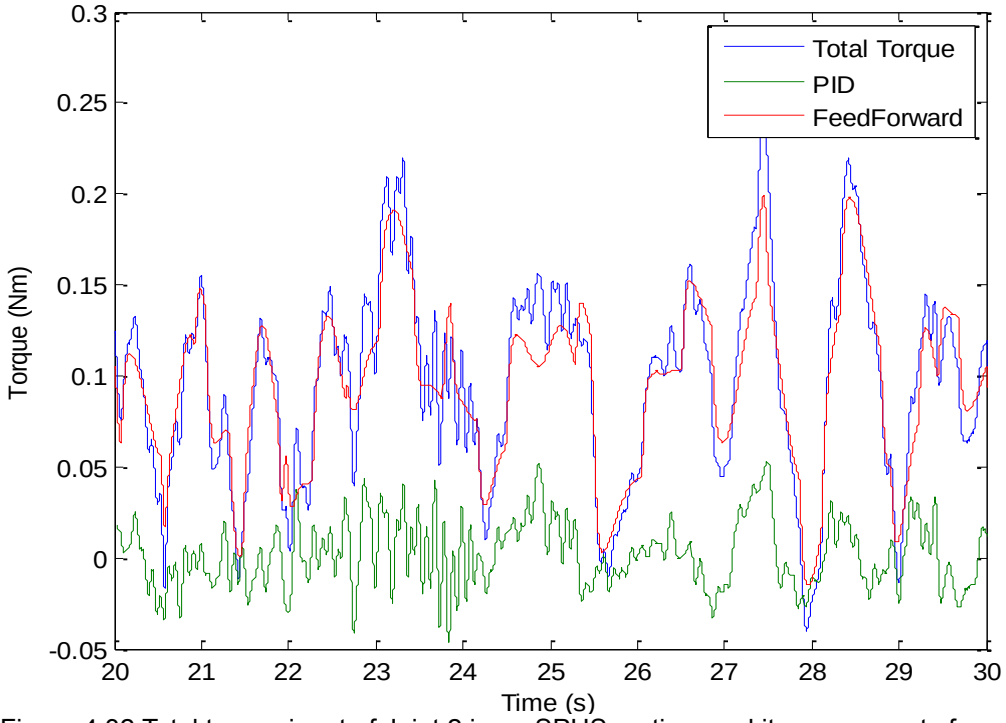

Figure 4.32 Total torque input of Joint 2 in an SPHS motion, and its components from feedforward model and PID controller.

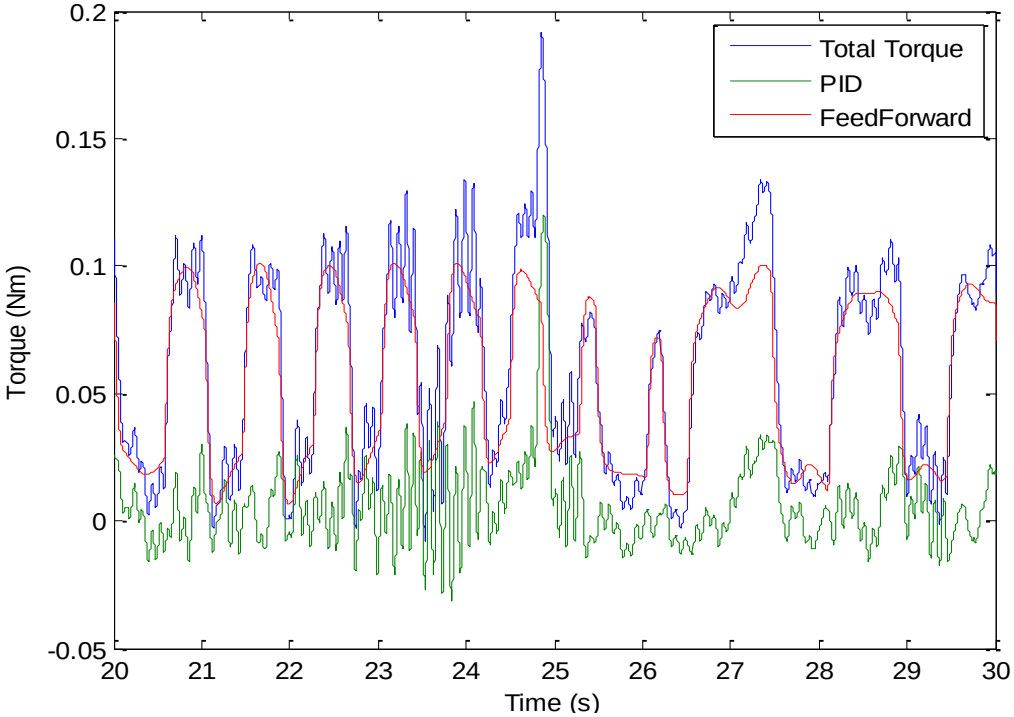

Figure 4.33 Total torque input of Joint 3 in an SPHS motion, and its components from feedforward model and PID controller.

Table 4.12 Correlation coefficient between total torque and each component

	FF	**PID**
Joint 2 Torque	0.9484	0.6080
Joint 3 Torque	0.9105	0.5762

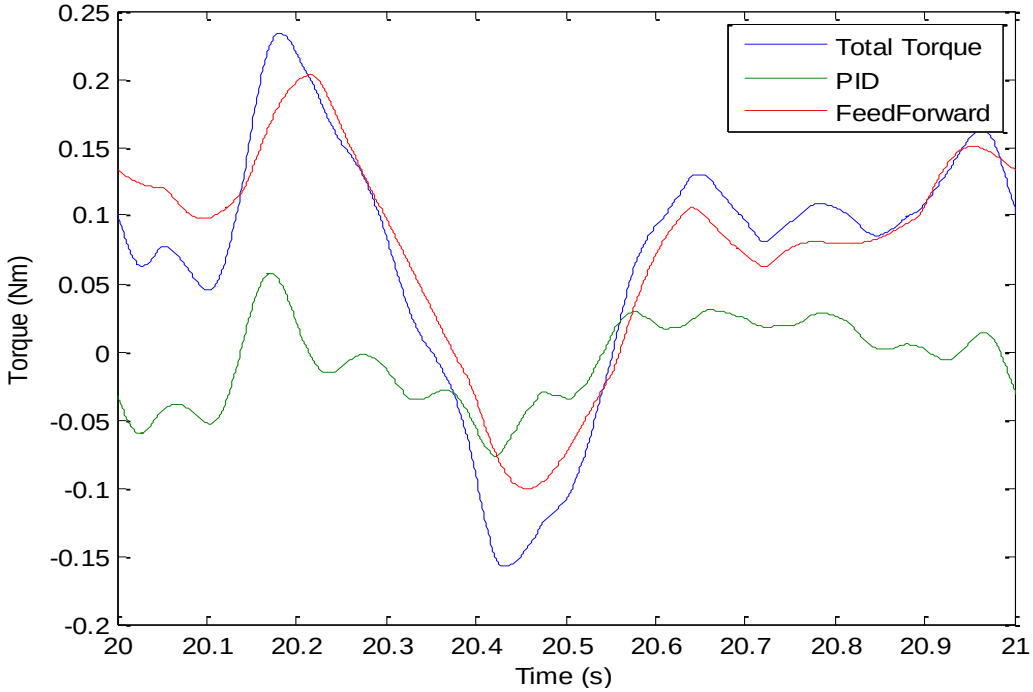

Figure 4.34 Total torque input of Joint 2 in a figure-of-eight motion, and its components from feedforward model and PID controller.

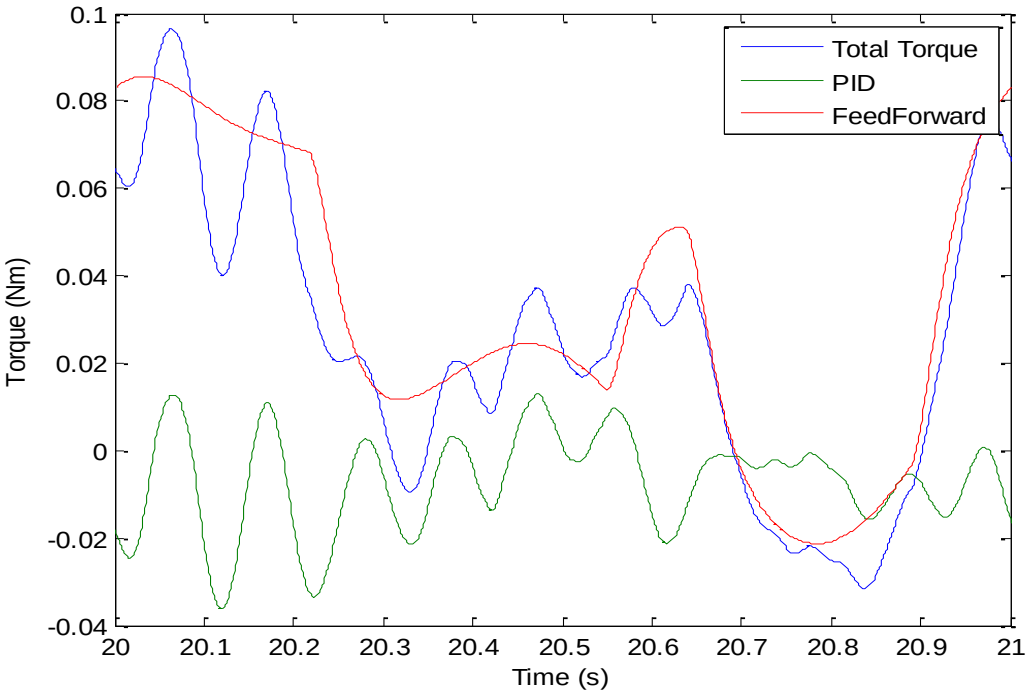

Figure 4.35 Total torque input of Joint 3 in a figure-of-eight motion, and its components from feedforward model and PID controller.

Table 4.13 Correlation coefficient between total torque and each component

	FF	**PID**
Joint 2 Torque	0.9550	0.6911
Joint 3 Torque	0.9433	0.1309

4.4 Concluding Remarks

In this chapter, system components and two cases of experimental set-ups are presented first, followed by the modelling of the robot system. Lastly, a new approach is proposed toward the design of robot excitation trajectories and its application to parameter estimation.

In this new method of designing optimal excitation trajectories for robot parameter estimation, SPHS signals have been utilised. The concept behind the method is based on synchronisation of the signals such that maximum information can be extracted from the system.

An SPHS signal has the advantage of being a low peak factor signal which contains only three parameters to optimise. This leads to a method with a reduced search space when compared to other traditional methods. In more detail, the SPHS signal for each of the robot's DOF is delayed to find minimum correlation trajectories, ensuring good coverage of the robot's workspace.

The major advantages of the method are twofold: it has fewer parameters to optimise compared to traditional methods such as the Fourier series; and it does not dwell on local minimum.

The proposed method is tested experimentally using the Omni robot. The model derived from the optimised trajectories shows a better joint torque prediction compared to a non-optimised one. The resulting models have also been tested and compared as part of a feedforward controller; the results compare positively for the optimised model. The controller with the optimised model shows a 4% improvement on position tracking.

Experiments are carried out using the above-estimated parameters and results are presented in the following chapters.

5 Initial Experiments on a Constrained Single Robot

In this chapter, results of the experimental applications of the extended inverse dynamics analysis procedure are given to illustrate the proposed technique. First, a single two-link manipulator carries out constrained motions while force tracking. The motions are designed to compare the performance of different controllers.

The chapter is organised as follows: Section 5.1 presents the experiment set-up in details; Section 5.2 presents the results in terms of figures and performance indicators; Section 5.3 presents the analysis of the results; Section 5.4 presents the conclusions.

5.1 Case I: One Robot Interacting with the Environment

The experiments start with the simplest possible case - Case I: one robot manipulates the load in a constrained motion. At least two DOF are needed to carry out both position and force control. The Omni robot is used which has two actuated links. The manipulator controller is developed so that it controls the force interaction with its environment.

The simplest type of active motion and force control on both moving and keeping contact is the end-effector sliding on a flat surface. The motion is relatively simple; however, it is not straightforward to mount a force sensor to accurately measure the forces perpendicular to the surface. Another possible solution for both force and position control is to have the manipulator connect with the environment through another link, and the link can rotate freely on the gripping point and the ground point. There is another advantage of this configuration, which is that the force sensor can be mounted on the link to measure reaction forces.

As shown in Figure 5.1, Robot 1 is controlled to rotate the load about a pivot point. The pivoting point is provided by Robot 2 with one of its wrist joints. Link

2-2 is clamped down on the robot stand so that the position of the pivot point is fixed.

The robot dynamic parameters obtained in Section 4.3 are used in the Dysim program and inverse dynamics models are generated for experimental use. For each experiment case, two Dysim models are produced: the extended inverse dynamics model and the conventional motion-only inverse dynamics model. They are used in the controllers (3.49), (3.50) and (3.52).

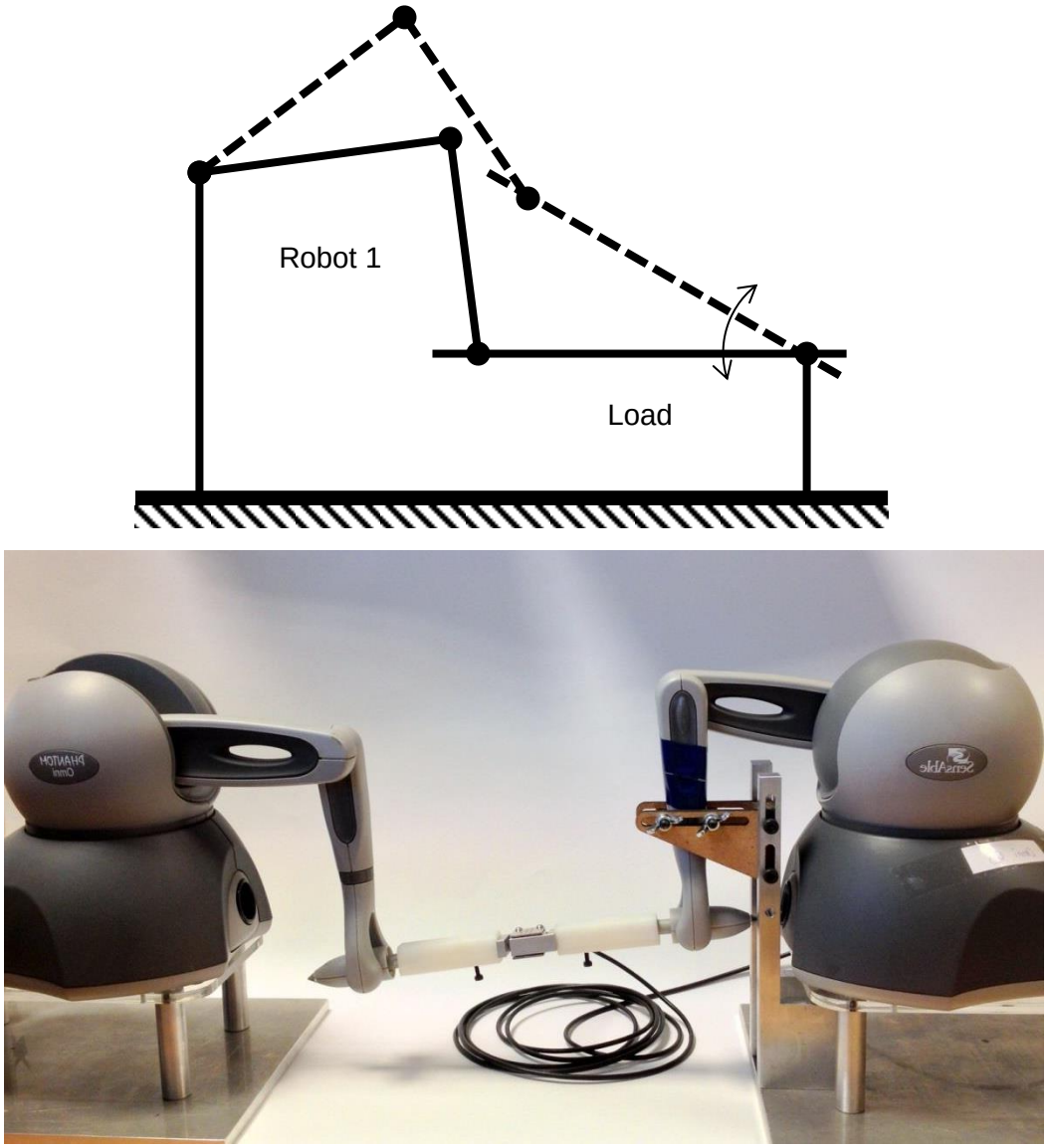

Figure 5.1 Case I: Schematic diagram (top) and actual rig (bottom). Robot 1 interacting with environment; Robot 2's arm is clamped down to the stand so that the wrist joint can be used as a free rotating pivot point

A set of motion demands is designed for the experiments. There are two major types of motion: smooth point-to-point motion and SPHS motion, a periodic multi-frequency sum of sine waves. The three motions are a slow point-to-point motion, a fast point-to-point motion and a fast SPHS motion.

The force demand is set to 0.5 N in the x_3 direction (Figure 4.12) throughout all experiments. It should be noted that 0.5 N is a relatively low value because for safety reasons it has to be set inside the continuous exertable force limit [108].

For experiment Case I, the motions are all demanded in the absolute angle of the load. Table 5.1 and Table 5.2 list the motion specifications. The time series of the demand signals are shown in Figure 5.5, Figure 5.10, and Figure 5.15.

Table 5.1 Motion specification for experiment Case I

Motion ID	**Signal Type**	**Duration/ period (s)**	**Coordinate applied**	**Initial position (deg)**	**Final position (deg)**
1	Point-to-point	2	Load angle θ_3	0	-20
2	Point-to-point	0.25	Load angle θ_3	0	-20
3	SPHS	2	Load angle θ_3	-10	-

Table 5.2 SPHS signal specification for experiment Case I

ω_0 (Hz)	**N_H**	**A_a**	**Range (deg)**	**Initial velocity (deg/s)**
0.5	10	0.0436	20	55.8137203641739

Figure 5.2 and Figure 5.3 show the Simulink models that run the experiments. In Figure 5.2, the inverse dynamics model is used to generate torques with given motion and force demands in advance of the main experiments. Note that the inputs of the Dysim Inverse block consist of both motion and force demands. The demand signals are saved in an array ready to be used in the main Simulink model. In Figure 5.3, the main Simulink model is shown, consisting of demand, controllers, outputs, and data logging. Figure 5.4 shows the subsystem for force interpretation and filtering. The input is the ADC values from the Omni Robot board; the output is force in newton. Force sensor calibration data from Section 4.1.1.3 is used here. The cut-off frequency for the low-pass filter is 50 rad/s.

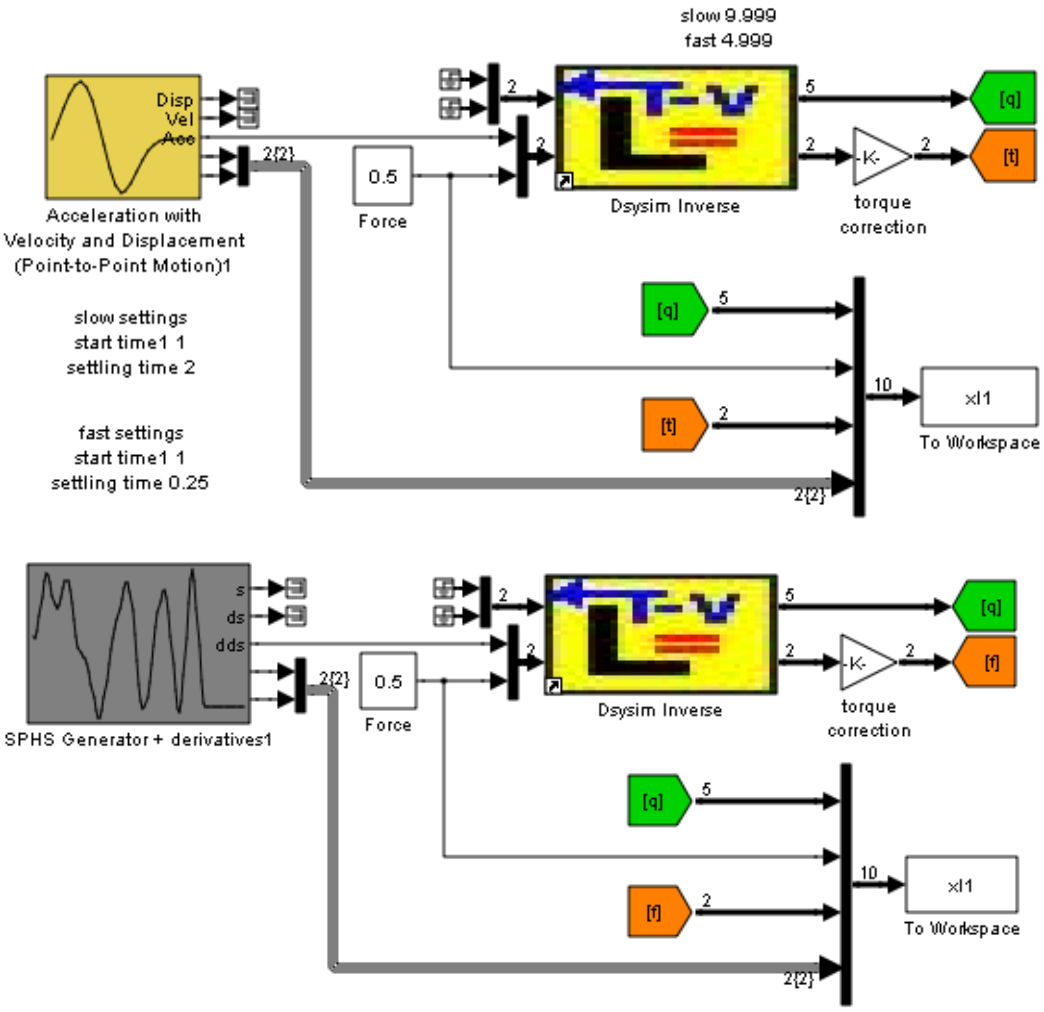

Figure 5.2 Simulink models of inverse dynamics to generate torques offline for the feedforward path and motion demand. Top: point-to-point motion. Bottom: SPHS motion

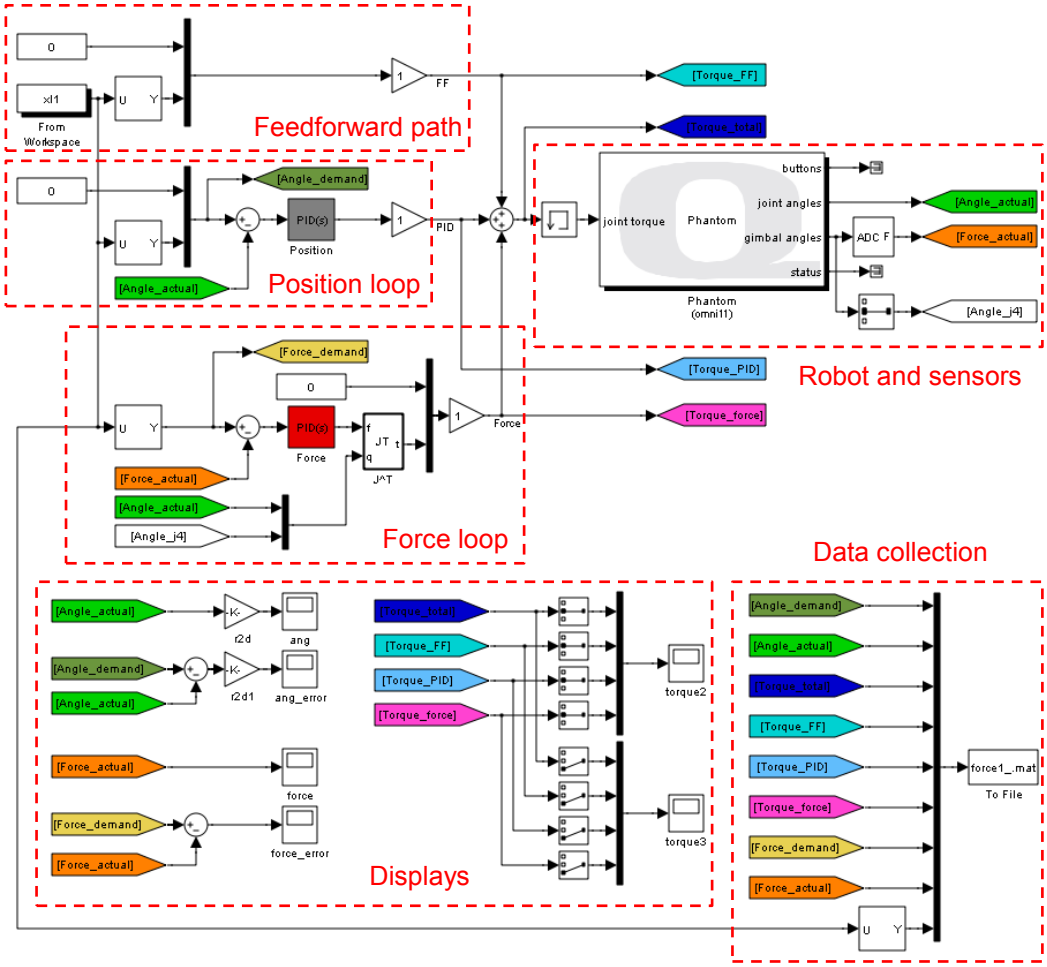

Figure 5.3 Simulink model of the control system and data collection

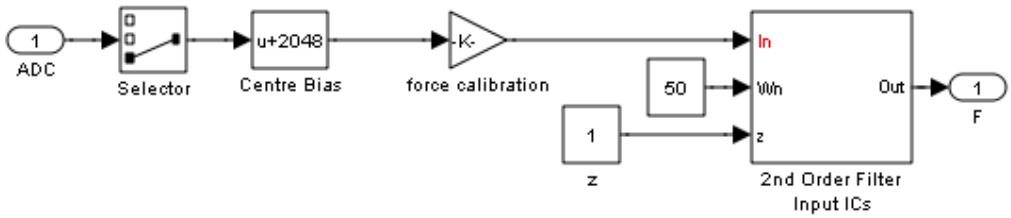

Figure 5.4 Force interpretation and filtering subsystem in Figure 5.3

PD plus gravity compensation is a popular position control scheme for manipulators [116]. It could be used on the Omni robots. However, the integral path is needed to overcome the steady state errors that are caused by the relatively high joint frictions, especially the stiction. Therefore, the PID controller is chosen for the joint position control.

The position and force PID controllers' gains are determined through separate experiments using the Tyreus-Luyben method. The Tyreus-Luyben tuning method is based on oscillations as in the classical Ziegler-Nichols' method, but with modified formulas for the controller parameters to obtain better stability in the control loop compared with the Ziegler-Nichols method [117].

The base joint is not concerned in the experiments so only a proportional gain of 1 is set. The PID gains for the remaining joints (motors) are listed in Table 5.3. The gains are adjusted slightly from the Tyreus-Luyben's results to ensure the position feedback loop and force feedback loop are stable. The filter coefficient (N) for the PID controller is set to 35. It determines the pole location of the filter in the derivative action.

The output of the force PID controller is transformed to joint torques with the transposed Jacobian matrix of the Robot. The torques are then added to the input on Motors 1-1 and 1-2 respectively, as shown in Figure 3.7. The reason to use two motors rather than just one motor to control the force is to avoid possible singular positions that will cause large torque inputs. For example, when Link 1-2 is aligned with Load, Motor 1-2 loses control of the force along Load's primary axis.

Table 5.3 Case I position loop and force loop PID gains

	$\mathbf{K_p}$	$\mathbf{K_i}$	$\mathbf{K_d}$
Motor 1-1	2.409	8.423	0.04971
Motor 1-2	0	0	0
Force	1	60.667	0.04946

All experiments are running at 1 kHz rate, *i.e.* using 0.001 fixed-step size in the Simulink configurations, and the solver ode1(Euler) is selected, if not stated otherwise.

Angles and force are then recorded and analysed. All motions are executed for 11 cycles so that data can later be analysed in a statistical manner. Data from the first cycle is discarded to avoid any transient effects there may be when the robots start to move from their resting positions to initial positions. All time series plots show only one of the cycles for clarity. The last cycle is chosen, arbitrarily, to be displayed. The results are presented in the following section.

5.2 Experimental Results

Time series of torque input components, angle errors and forces for each motion type are presented in this section, as well as their error distributions. The distributions of load angle errors and internal force errors in different controllers are also plotted, using the 'boxplot' command in Matlab. On each box, the central red mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually [118]. Numerical results are also reported.

5.2.1 Time Series and Error Distributions

The figures are gathered into three groups by motion type:

- Slow point-to-point motion
- Fast point-to-point motion
- SPHS motion

In each motion type, there are five figures, in the following order:

- Angle demand and errors
- Angle error distributions
- Force demand and actual force
- Force error distributions
- Joint torque components

Table 5.4 Table of figures presented in Case I

	Slow point-to-point	**Fast point-to-point**	**SPHS**
Angle demand and errors	Figure 5.5	Figure 5.10	Figure 5.15
Angle error distributions	Figure 5.6	Figure 5.11	Figure 5.16
Demand and actual force	Figure 5.7	Figure 5.12	Figure 5.17
Force error distributions	Figure 5.8	Figure 5.13	Figure 5.18
Joint torque components	Figure 5.9	Figure 5.14	Figure 5.19

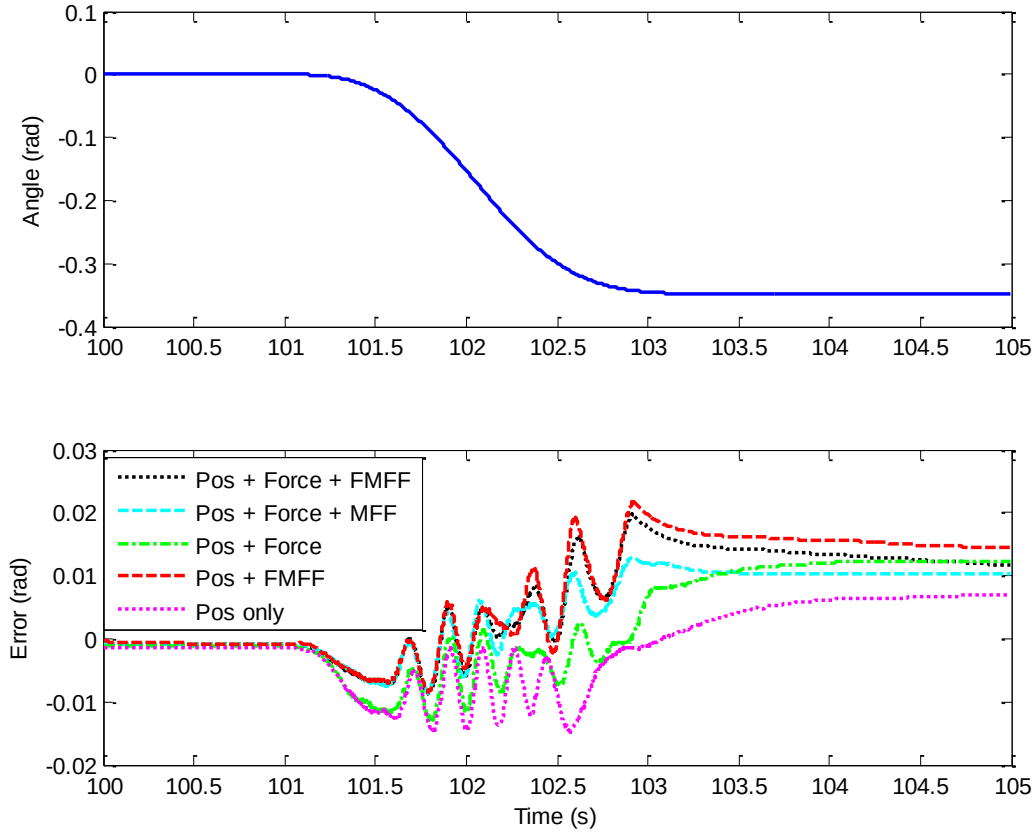

Figure 5.5 Case I experiment results: load angle in slow point-to-point motion. Top: angle demand. Bottom: angle errors in different controllers.

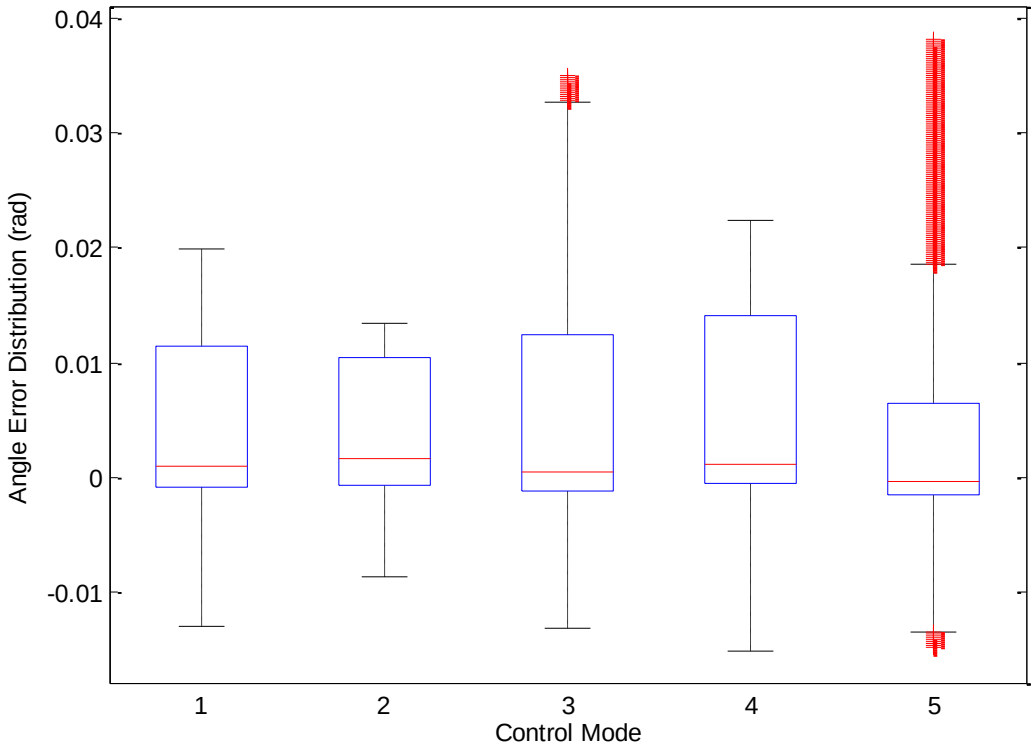

Figure 5.6 Case I experiment results: load angle error distribution in slow point-to-point motion.

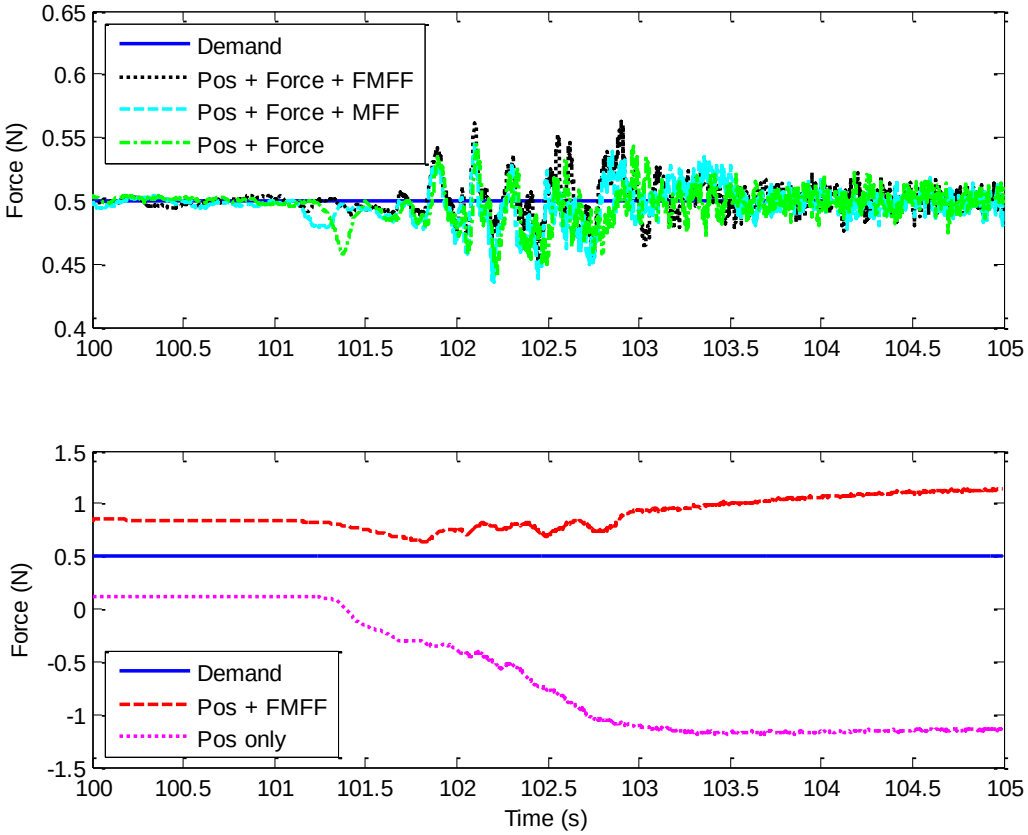

Figure 5.7 Case I experiment results: internal force in slow point-to-point motion. Top: controllers with force feedback. Bottom: controllers without force feedback.

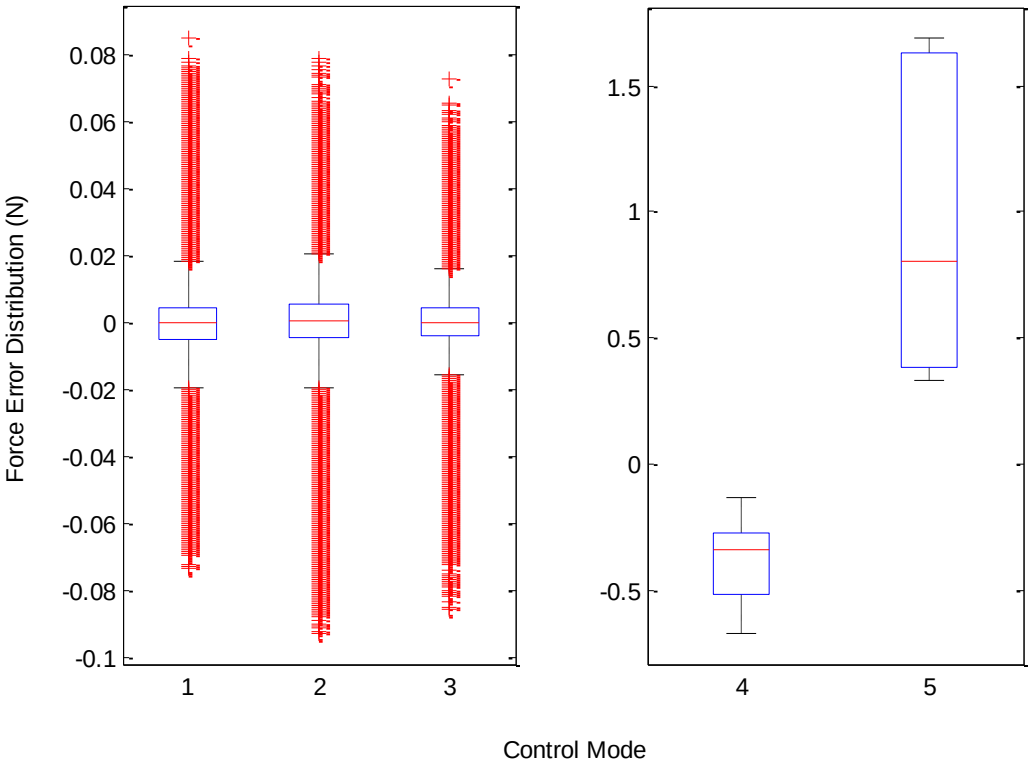

Figure 5.8 Case I experiment results: internal force error distribution in slow point-to-point motion.

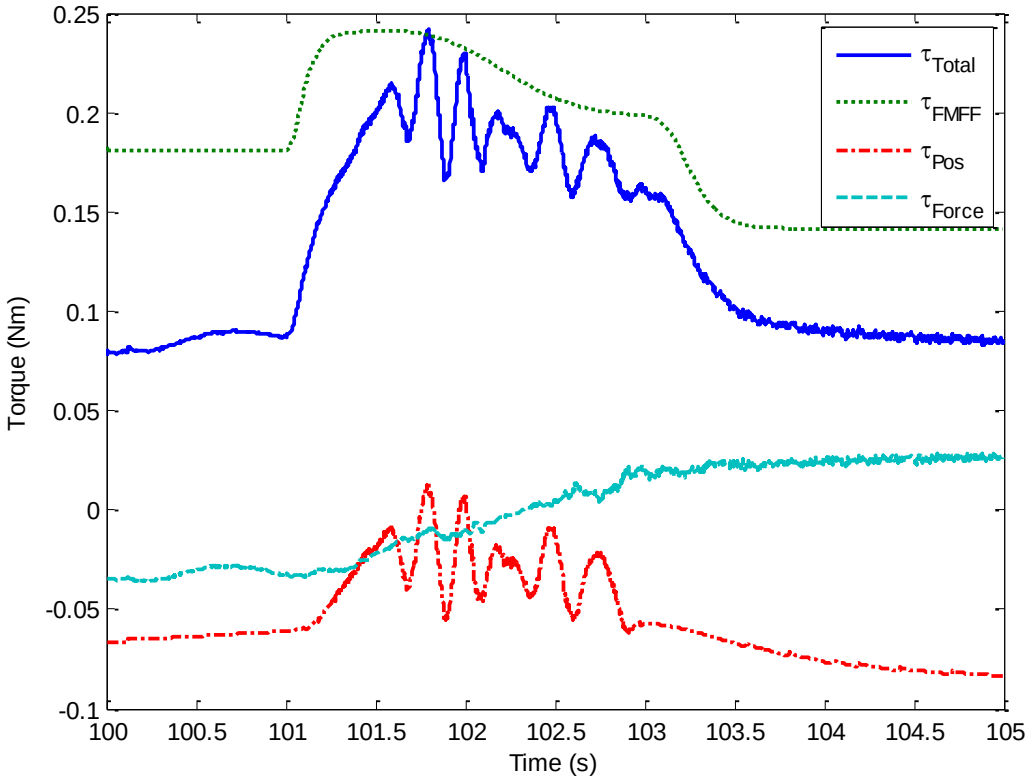

Figure 5.9 Case I experiment results: Controller 1 Joint 1 torque inputs in slow point-to-point motion.

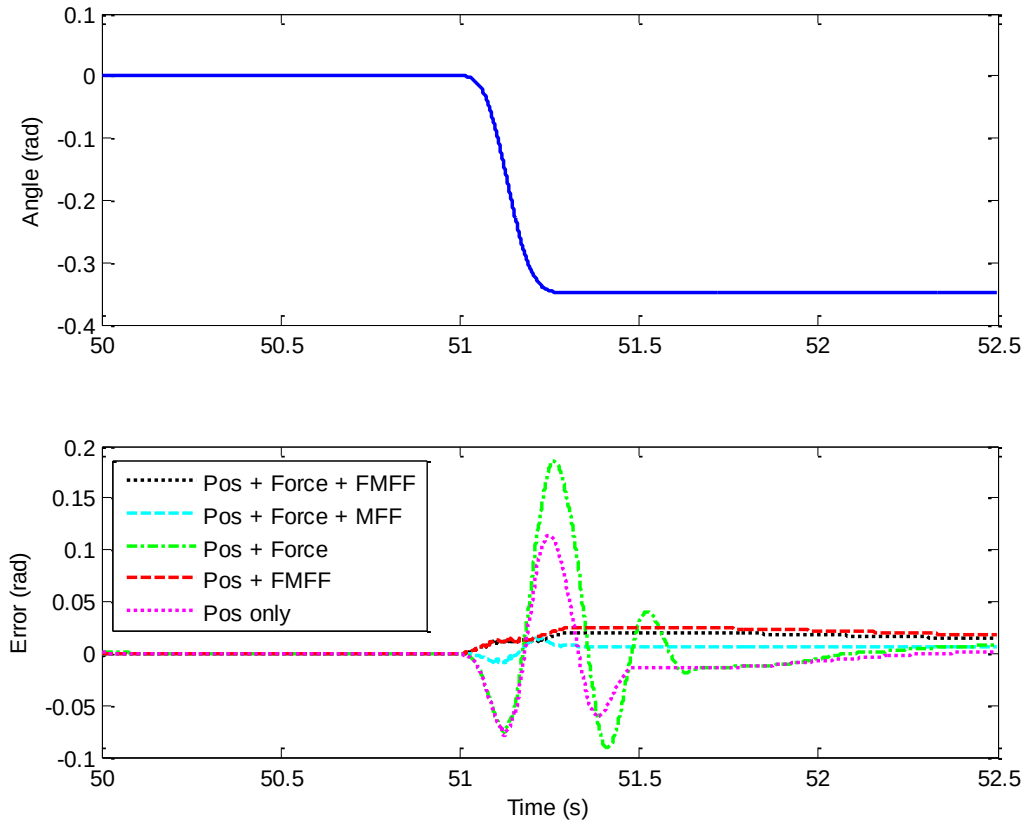

Figure 5.10 Case I experiment results: load angle in fast point-to-point motion. Top: angle demand. Bottom: angle errors of different controllers.

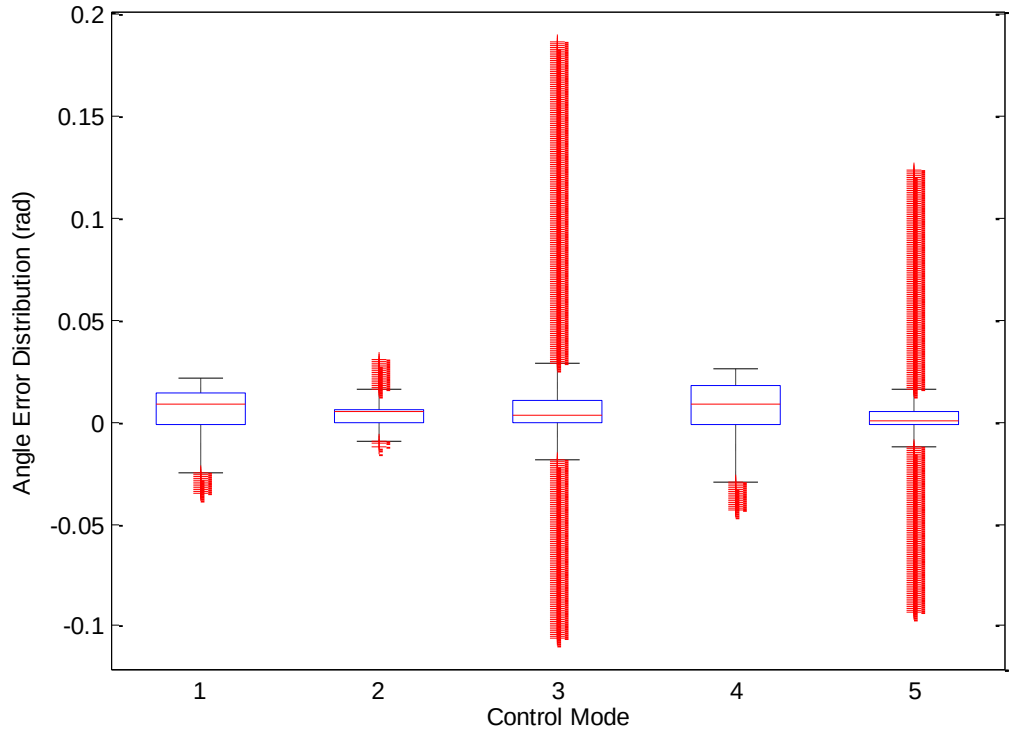

Figure 5.11 Case I experiment results: load angle error distribution in fast point-to-point motion.

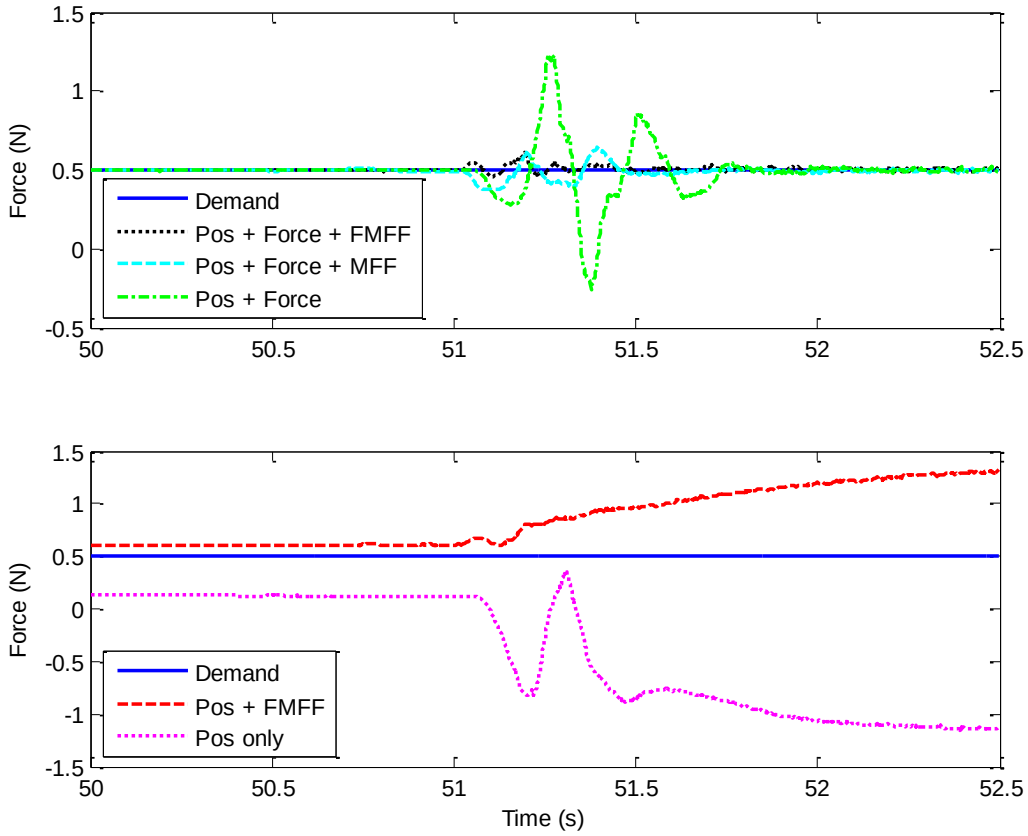

Figure 5.12 Case I experiment results: internal force in fast point-to-point motion. Top: controllers with force feedback. Bottom: controllers without force feedback.

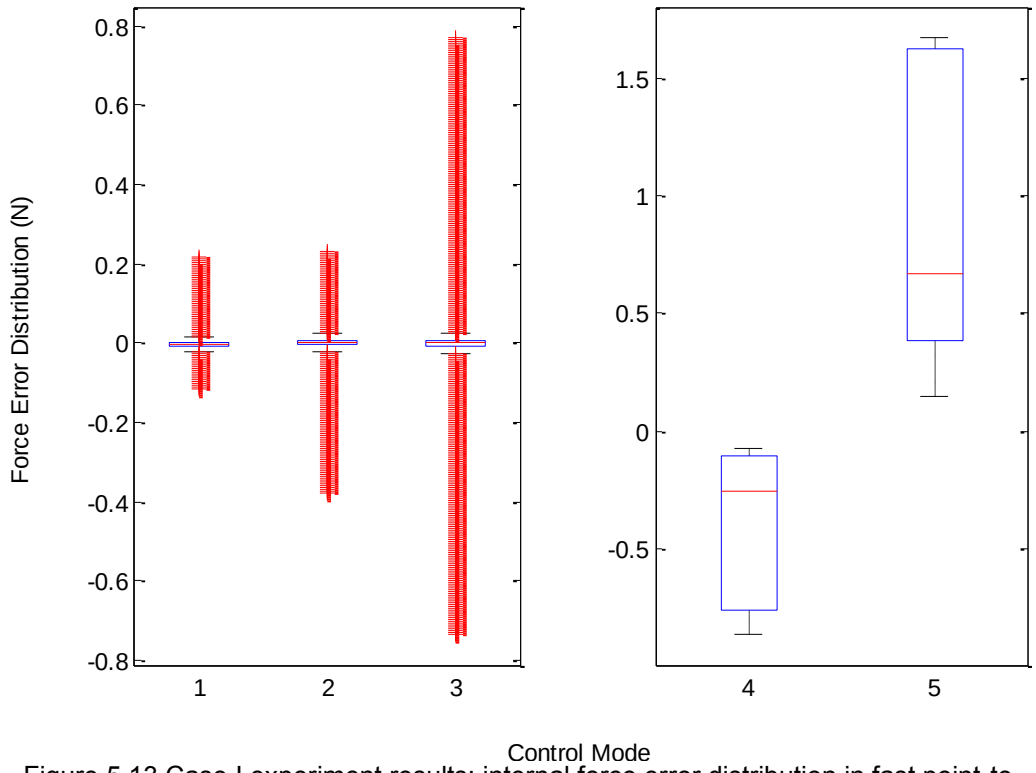

Figure 5.13 Case I experiment results: internal force error distribution in fast point-to-point motion. 1.Pos + Force + FMFF; 2.Pos + Force + MFF; 3.Pos + Force; 4.Pos + FMFF; 5.Pos only

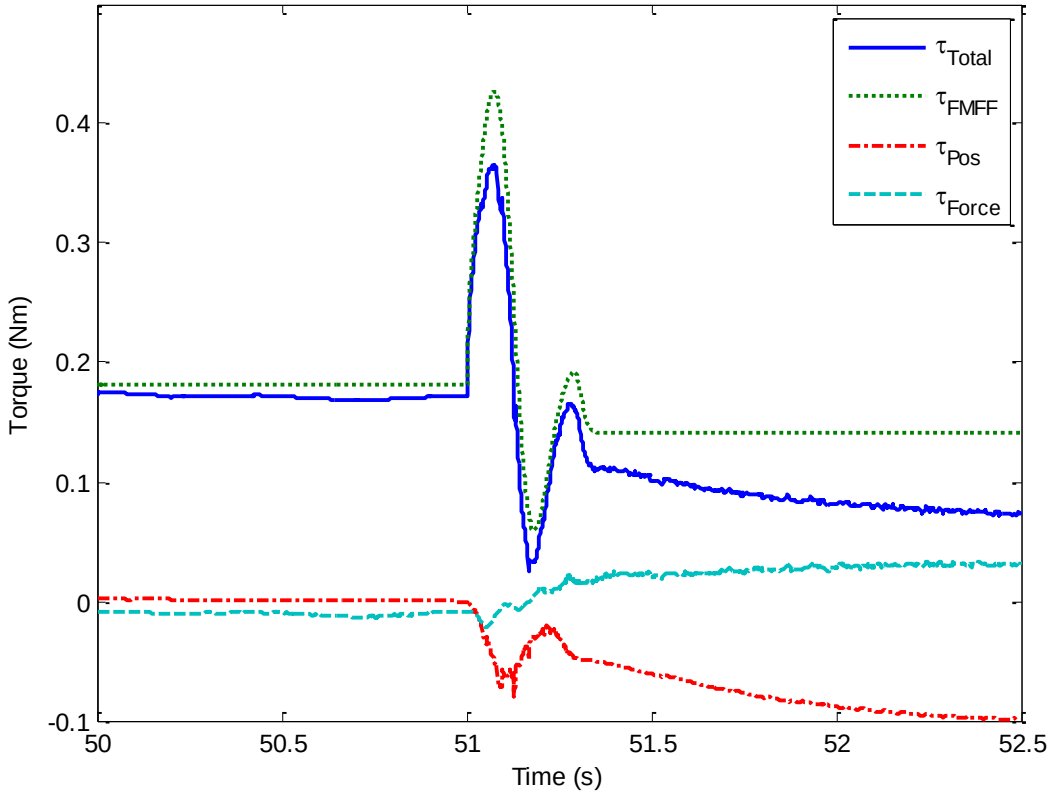

Figure 5.14 Case I experiment results: Controller 1 Joint 1 torque inputs in fast point-to-point motion.

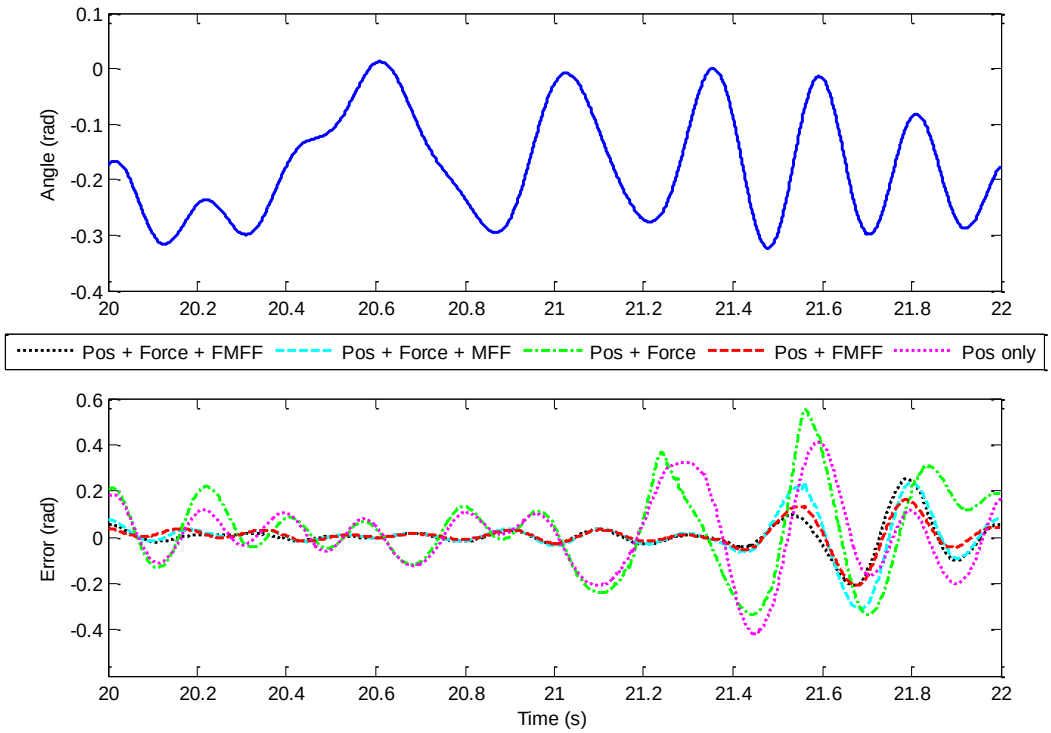

Figure 5.15 Case I experiment results: load angle in SPHS motion.
Top: angle demand. Bottom: angle errors of different controllers.

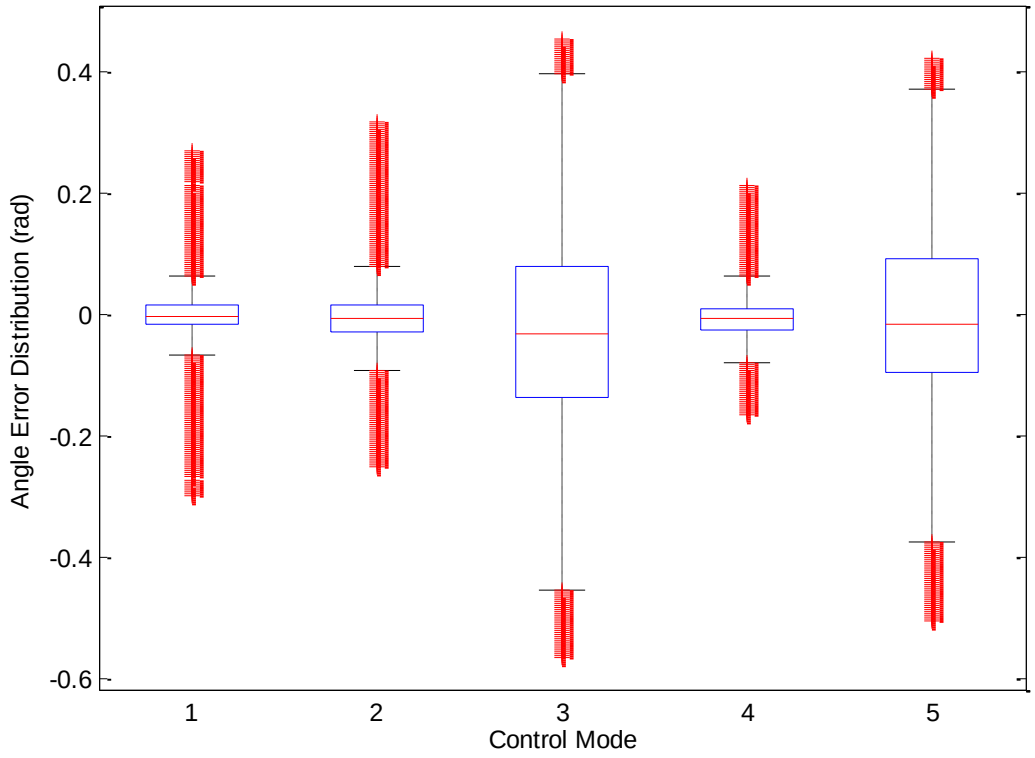

Figure 5.16 Case I experiment results: load angle error distribution in SPHS motion.
1.Pos + Force + FMFF; 2.Pos + Force + MFF; 3.Pos + Force; 4.Pos + FMFF; 5.Pos only

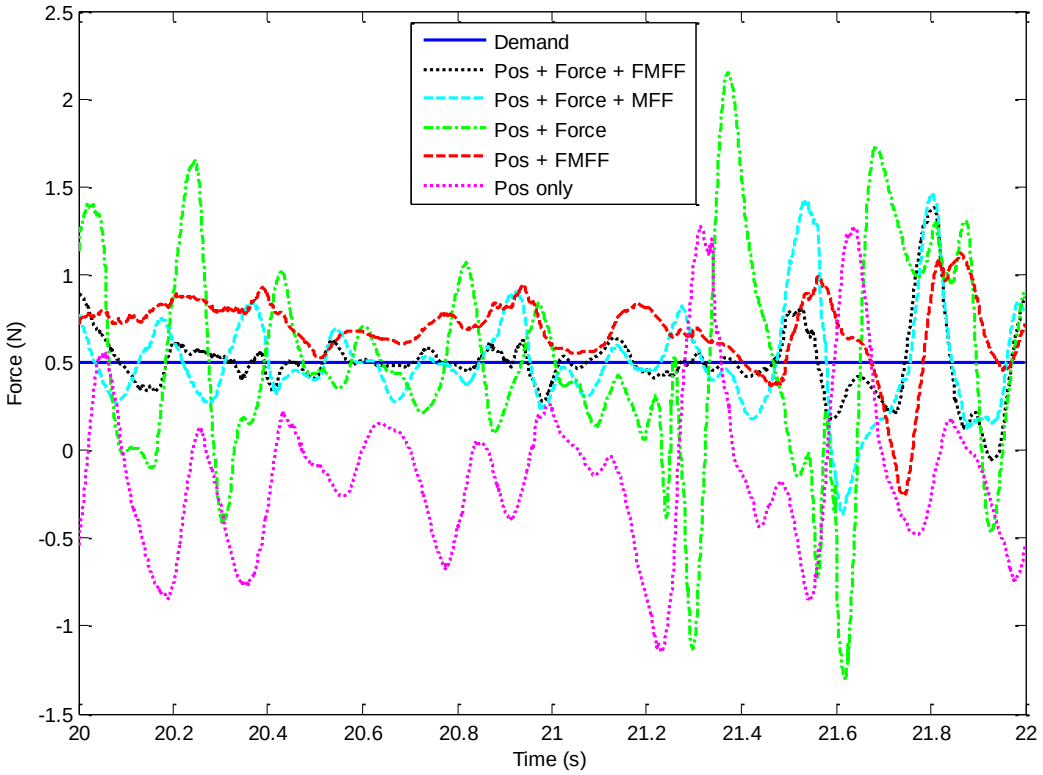

Figure 5.17 Case I experiment results: internal force in SPHS motion.

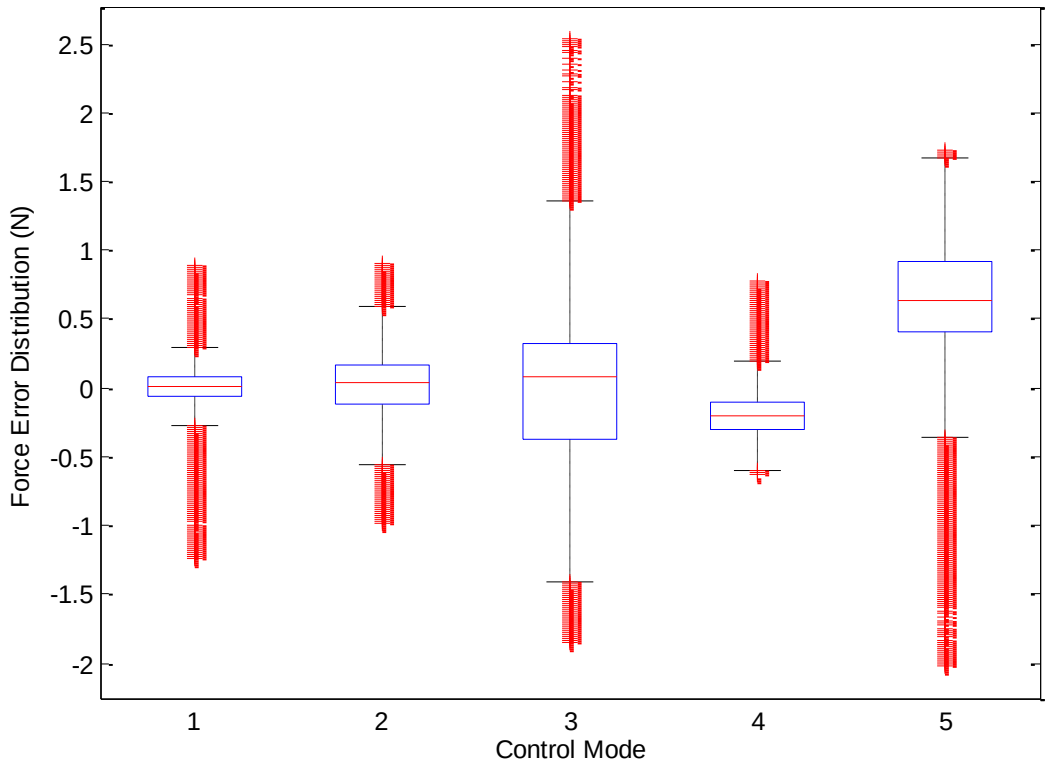

Figure 5.18 Case I experiment results: internal force error distribution in SPHS motion.
1.Pos + Force + FMFF; 2.Pos + Force + MFF; 3.Pos + Force; 4.Pos + FMFF; 5.Pos only

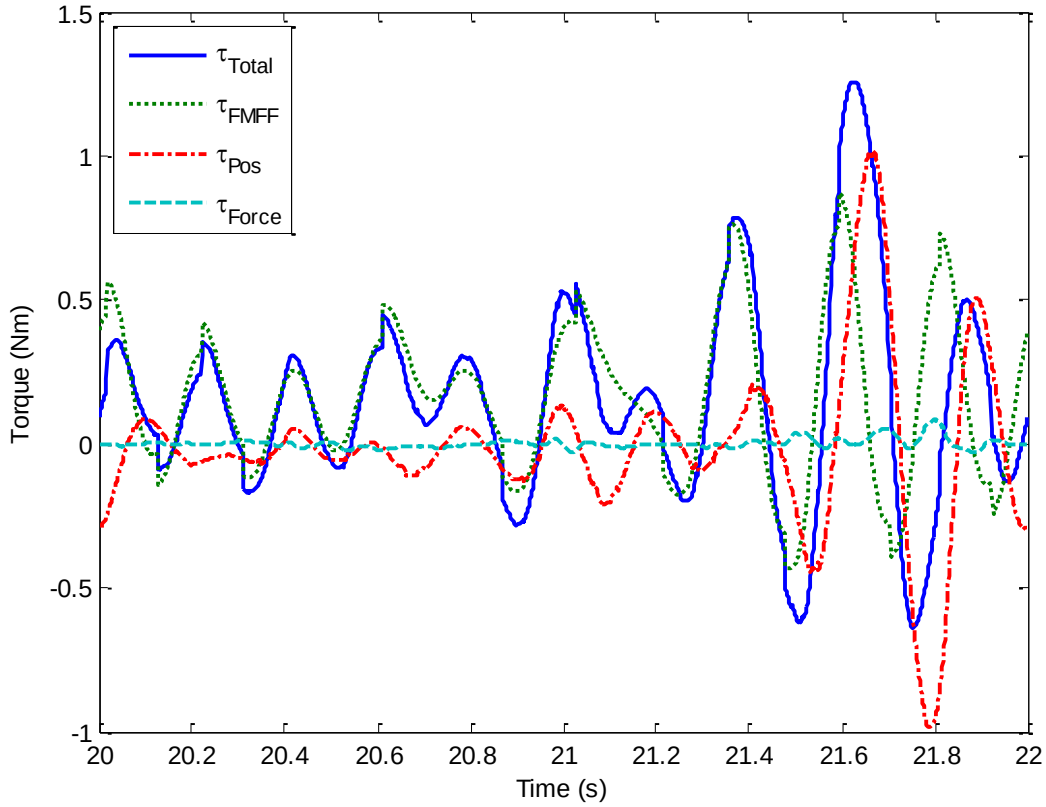

Figure 5.19 Case I experiment results: Controller 1 Joint 1 torque inputs in SPHS motion.

5.2.2 Numerical Results

The performances of the controllers are measured by the root mean square errors (RMSE) of the variables. Table 5.5 to Table 5.7 list the position errors and force errors for different controllers from different motions. The error values in the tables are averaged over ten cycles for each motion. The highlighted cells denote the smallest errors among different controllers.

Table 5.5 Position and force errors of *slow point-to-point* motion

Controller		Position RMSE (rad)	Force RMSE (N)
1	Pos + Force + FMFF	7.594×10^{-3}	1.796×10^{-2}
2	Pos + Force + MFF	5.800×10^{-3}	2.041×10^{-2}
3	Pos + Force	6.102×10^{-3}	1.992×10^{-2}
4	Pos + FMFF	8.470×10^{-3}	2.794×10^{-1}
5	Pos only	8.312×10^{-3}	1.016×10^{0}

Table 5.6 Position and force errors of *fast point-to-point* motion

Controller		Position RMSE (rad)	Force RMSE (N)
1	Pos + Force + FMFF	1.060×10^{-2}	4.035×10^{-2}
2	Pos + Force + MFF	7.378×10^{-3}	8.535×10^{-2}
3	Pos + Force	7.229×10^{-2}	1.684×10^{-1}
4	Pos + FMFF	1.207×10^{-2}	1.977×10^{-1}
5	Pos only	5.792×10^{-2}	8.475×10^{-1}

Table 5.7 Position and force errors of *SPHS* motion

Controller		Position RMSE (rad)	Force RMSE (N)
1	Pos + Force + FMFF	6.780×10^{-2}	2.163×10^{-1}
2	Pos + Force + MFF	8.571×10^{-2}	2.963×10^{-1}
3	Pos + Force	1.818×10^{-1}	5.960×10^{-1}
4	Pos + FMFF	5.506×10^{-2}	2.737×10^{-1}
5	Pos only	1.637×10^{-1}	7.967×10^{-1}

5.3 Analysis and Comparison

In general, the faster the motion, the better performance the feedforward controllers deliver compared with the other controllers. This is because at lower speed, the system appears to be linear and the basic PID controller works well near the linearization points, but at higher speed, the nonlinearity becomes dominant so the PID controllers cannot cope. The nonlinearity is a result of system dynamics.

Amongst all motions, the best force tracking is consistently achieved by Controller 1.

Take Controller 3's performance as a benchmark. In slow point-to-point motion, Controller 1 improves force tracking but worsens position tracking. In contrast, Controller 2 improves position tracking but worsens force tracking. In fast point-to-point and SPHS motions, both Controllers 1 and 2 improve position and force tracking.

When comparing Controllers 1 and 2, the former performs better in SPHS motion for both position and force tracking, and in point-to-point motions for force tracking; and the latter only performs better in point-to-point motions for position tracking.

It is noticed that in the SPHS motion, Controller 4 performs better than Controllers 2 and 3 in both position and force tracking. This means that the force feedforward controller works better without force feedback loop.

It is clear that Controller 5 performs the worst in force tracking due to its lack of any form of force control. It is also interesting to notice that, when comparing Controller 3 with Controller 5, one would expect Controller 3 to produce worse position tracking with added force feedback loop, which is the fact in the fast point-to-point motion and SPHS motion. However, counter-intuitively, Controller 3 improves position tracking in the slow point-to-point motion.

Figure 5.9, Figure 5.14, and Figure 5.19 reveal the torque inputs of Joint 1 and the contributions from different control loops. They all show that the feedforward loop makes up the most of the input signals.

5.4 Concluding Remarks

This chapter presents the details of the experimental set-up from the initial experiments of Case I. A two-DOF robot is carrying a bar-shaped load at one end. The other end of the load is fixed to ground but can rotate freely. Different motions on load angle are designed for the experiments: slow point-to-point motion, fast point-to-point motion, and SPHS motion. The results are then reported and analysed. Results show that Controller 1 performs the best force tracking in all the three types of motion. In addition, the controllers with feedforward loop perform better at higher speed motions, *i.e.* the fast point-to-point motion and SPHS motion.

The results confirm the performance of the proposed controller hence further experiment can be carried out on cases that are more complicated. In the next chapter, the second robot is introduced and experiments with cooperating robots are carried out.

6 Experiments on Cooperative Robots

The previous chapter reports the set-up and results of experiment Case I: the single robot moving under constraint. The results for the proposed extended inverse dynamics controller were promising. This chapter presents further experiments using the method for collaborative robots.

In this chapter, detailed set-ups are described and results of the experimental Case II are given. First, a second manipulator (Robot 2) is introduced to hold the position of the pivoting point on the load and Robot 1 carries out the same constrained motions as in Case I. Then both robots are commanded to move the load while tracking internal force. These results were also published in [119]

The chapter is organised as follows: Section 6.1 presents the experiment set-up in details for Case II; Section 6.2 presents the results in terms of figures and performance indicators; Section 6.3 presents the analysis of the results; Section 6.4 concludes this chapter.

6.1 Case II: Two Robots Cooperating

In Case II, two experiments are designed, Case II-a and Case II-b. In Case II-a (Figure 6.1), following the experiments on Case I, the fixture on Robot 2 is removed to release the Link 2-2. The Robot 2 is separately commanded to hold the position of the load as if it is still fixed. Then the same experiments from Case I are repeated. In Case II-b (Figure 6.5), Robot 2 is included in the feedforward model and both robots move the load cooperatively with internal force tracking.

The same dynamic parameters are used for Robot 2 in the Dysim program and inverse dynamics models are generated for experimental use.

The base joints are not concerned in the experiments so only a proportional gain of 1 is set. The PID gains for the remaining joints (motors) are the same as in Case I and listed in Table 6.1. The filter coefficient (N) for the PID controller is set to 65. It determines the pole location of the filter in the derivative action.

The output of the force PID controller is transformed to joint torques with the transposed Jacobian matrix of the Robot. The torques are then added to the input on Motors 1-1 and 1-2 respectively, as shown in Figure 3.7.

Table 6.1 Case II position loop and force loop PID gains

	K_p	K_i	K_d
Motor 1-1	2.409	16.846	0.0994
Motor 1-2	0	0	0
Motor 2-1	2.409	16.846	0.0994
Motor 2-2	2.091	5.867	0.1075
Force	1	12.133	0.05

6.1.1 Case II-a: One Moving Robot and One Static Robot

In this case, the fixture used to hold the arm of Robot 2 is removed, thus active control is required to maintain the position. Instead, Robot 2 is programmed to hold the position with PID controllers on the actuated joints as shown in Figure 6.1. The motion of the load remains the same as in Case I.

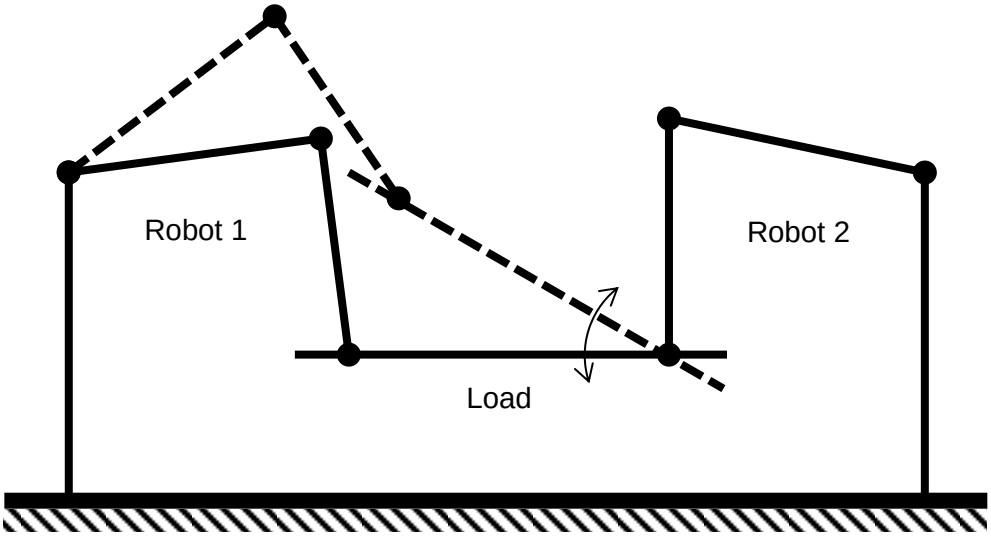

Figure 6.1 Case II-a: Schematic diagram. Robot 1 moving the load while Robot 2 holds position

For experiment Case II-a, the motions are all defined on the absolute angle of the load. Same motions from Case I are used: slow and fast point-to-point motions, and SPHS motions. The motion specifications are listed in Table 5.1 and Table 5.2.

Since the motions are the same as in Case I, the Simulink models used to generate the feedforward torques are the same as shown in Figure 5.2. However, the main program is slightly different as shown in Figure 6.2. In addition to demand signals, controllers, outputs, and data logging, there is a separate position controller for Robot 2 in the top right corner of the screenshot. Figure 6.3 shows the subsystem for the Robot 2 position control.

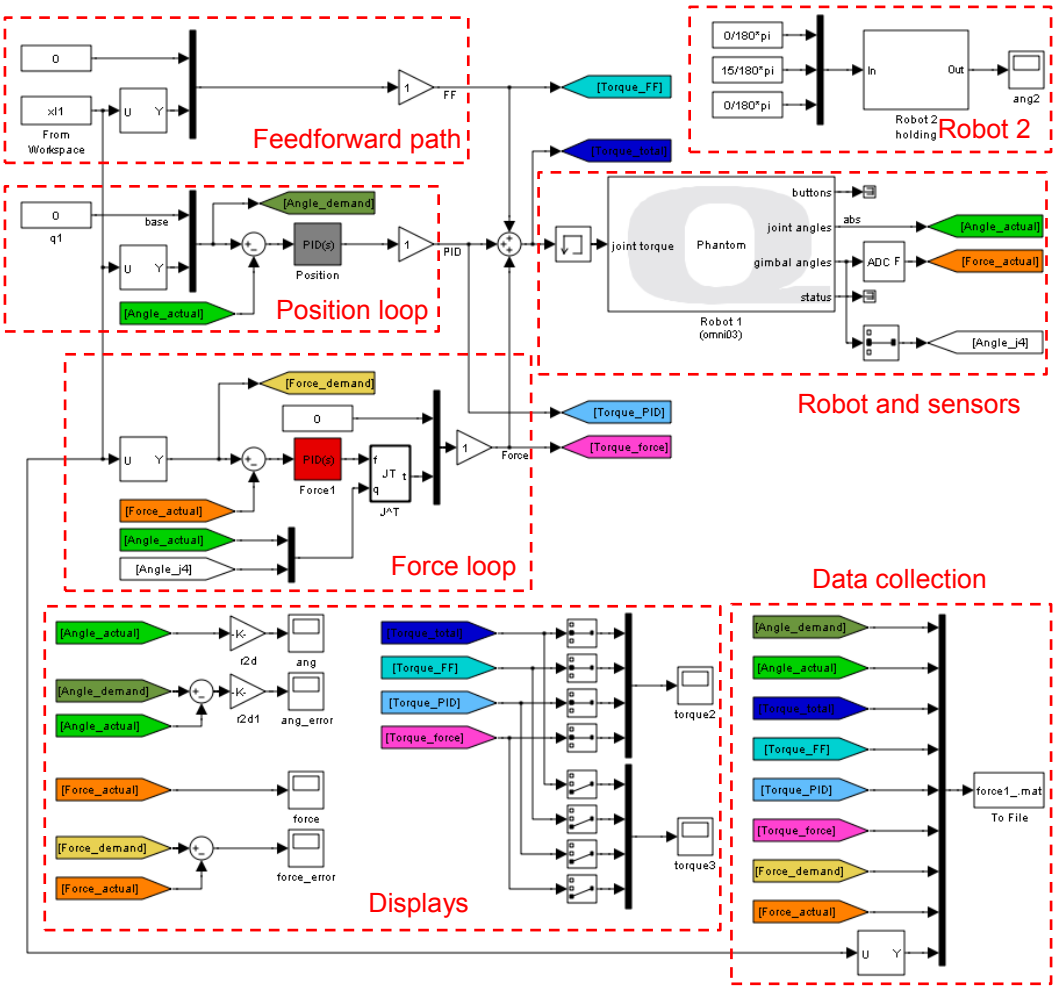

Figure 6.2 Simulink model of the control system and data collection

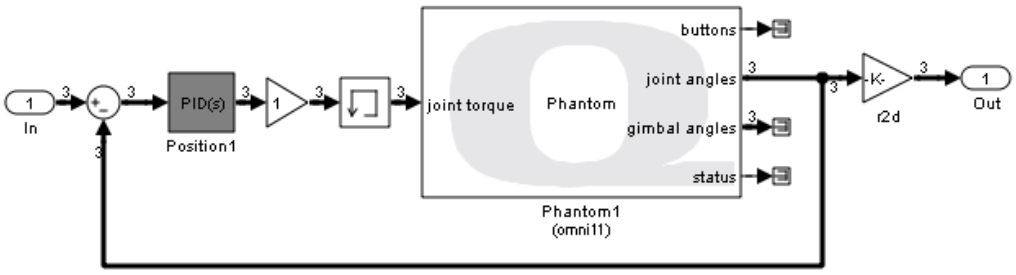

Figure 6.3 Subsystem in Figure 6.2 for Robot 2 position control

6.1.2 Case II-b: Two Moving Robots

In this case, both of the robots are moving the load together. Both robot models are used in the new controller. In other words, the robots are now actively cooperating with each other to achieve both motion and force tracking (Figure 6.5).

Figure 6.4 Case II-b: the actual rig. Two robots moving a load with force sensor

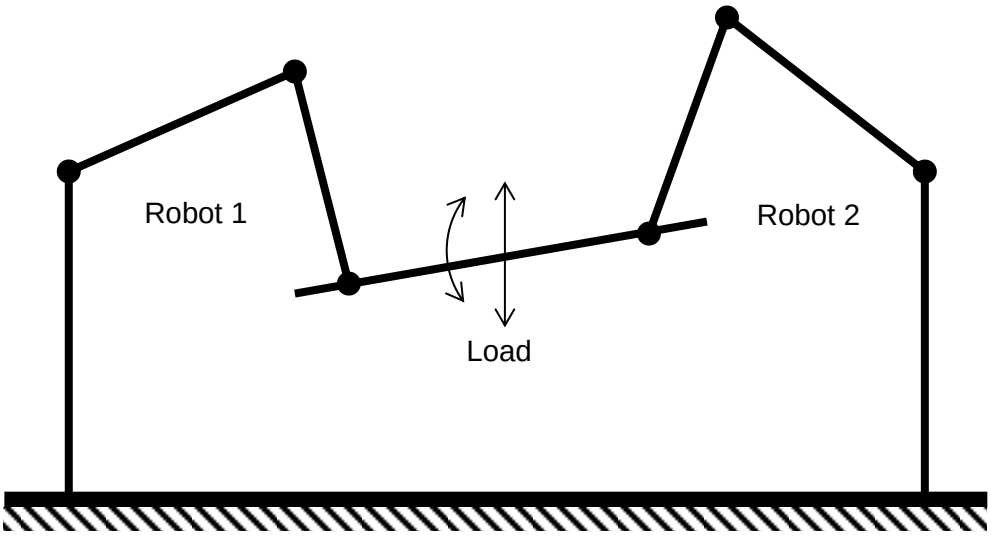

Figure 6.5 Case II-b: Schematic diagram. Two robots moving a load with force sensor

Due to the introduction of the second robot, there are now four actuators. Moreover, because of the connection made through the load, there are three motion DOF left in the system. Therefore, new motions are designed for two robots' cooperation. For experiment Case II-b, the motions are demanded on the position and absolute angle of the load (Figure 4.12).

For point-to-point motions, different motions with the same duration are applied to the load global axes x and z, and absolute load angle. Table 6.2 lists the motion specifications. The time series of the demand signals are shown in Figure 6.24 and Figure 6.29.

Table 6.2 Motion specifications for experiment Case II-b point-to-point motion

Motion ID	Signal Type	Duration/ period (s)	Coordinate applied	Initial positions	Final positions
4	Point-to-point	2	Load angle θ_3	15 (deg)	-5 (deg)
			Load x axis	0.22 (m)	0.27 (m)
			Load z axis	-0.1 (m)	0 (m)
5	Point-to-point	0.4	Load angle θ_3	15 (deg)	-5 (deg)
			Load x axis	0.22 (m)	0.27 (m)
			Load z axis	-0.1 (m)	0 (m)

For SPHS motion, it is not practical to apply SPHS signals to all three DOF due to the limited workspace. The links, especially the Load, would hit physical limits, *i.e.* the Load hitting the robot base. Therefore, the SPHS signal is applied only to the Load angle. On the z-axis, two identical point-to-point motions are applied with opposite directions, so the Load goes up then comes down after a certain amount of time. The Load remains the same position in x-axis. Consequently, in the combined motion the Load should move up and down while rotating around its COG. Table 6.3 and Table 6.4 list the motion specifications. The time series of the demand signals are shown in Figure 6.34

Table 6.3 Motion specifications for experiment Case II-b combined motion

Motion ID	Signal Type	Duration/ period (s)	Coordinate applied	Initial positions	Final positions
6	SPHS	5	Load angle θ_3	15 (deg)	-
	Point-to-point	5	Load x axis	0.25 (m)	0.25 (m)
	Point-to-point	5	Load z axis	-0.1 (m)	0 (m)

Figure 6.6 and Figure 6.7 show the Simulink models that are used to run the experiments.

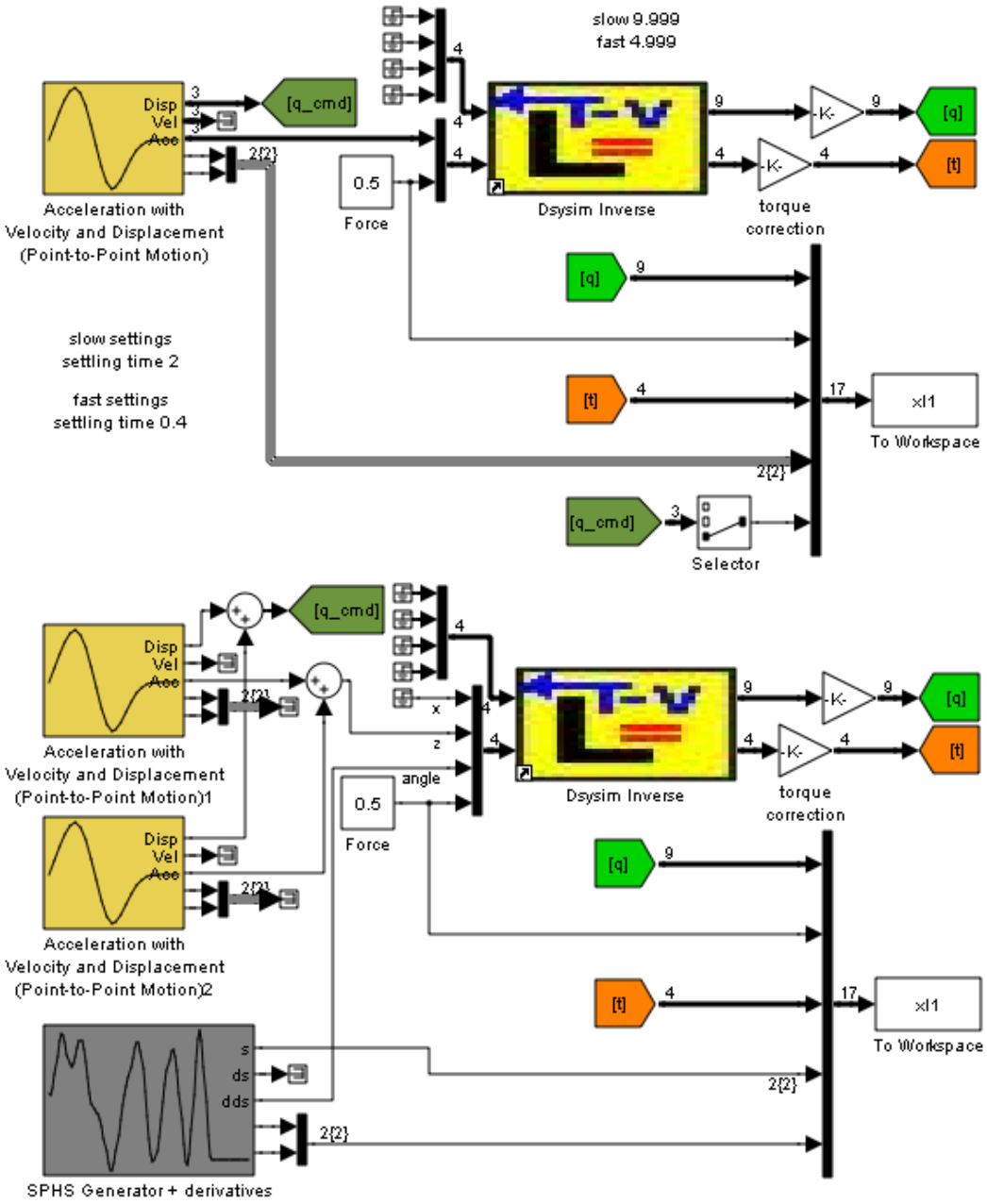

Figure 6.6 Simulink models of inverse dynamics to generate torques offline for the feedforward path and motion demand. Top: point-to-point motion. Bottom: combined SPHS motion

Table 6.4 SPHS signal specification for experiment Case II-b

ω_0 (Hz)	N_H	A_a	Range (deg)	Initial velocity (deg/s)
0.2	10	0.0654	30	33.8685379465

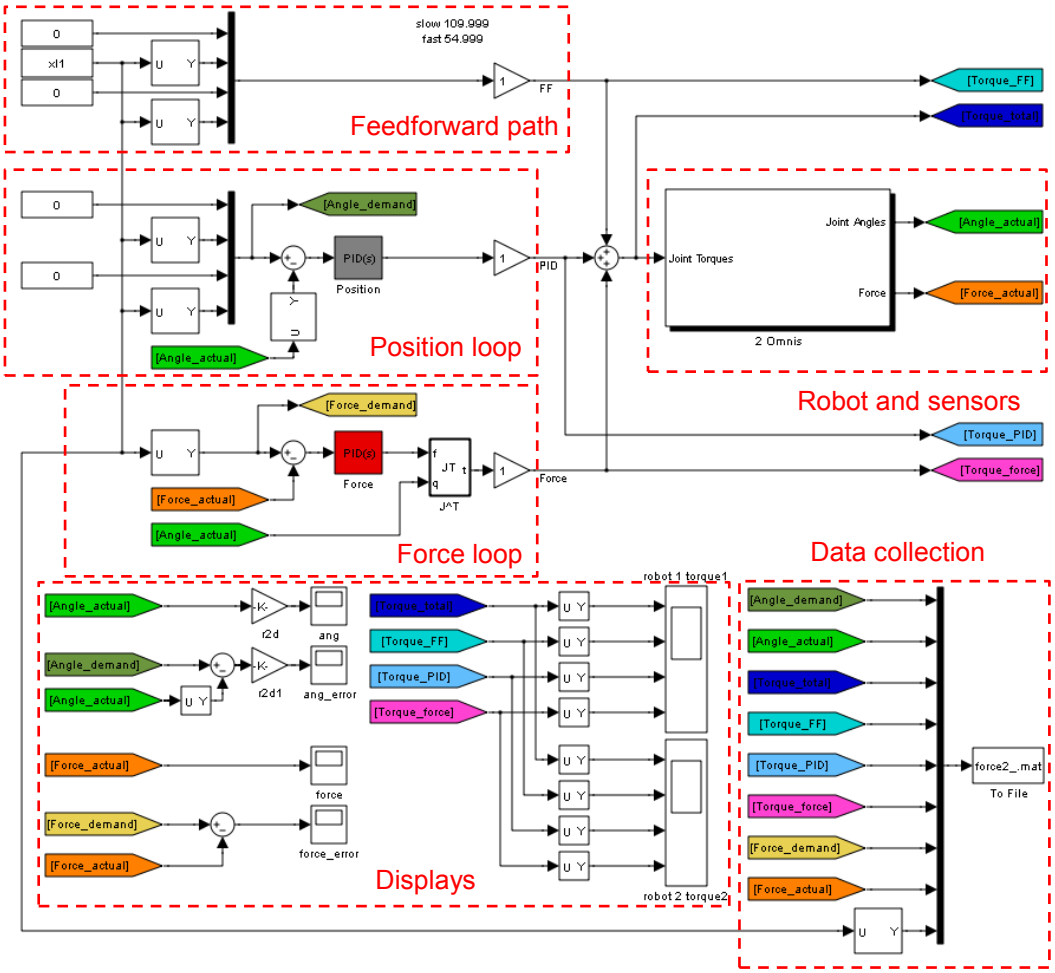

Figure 6.7 Simulink model of the control system and data collection

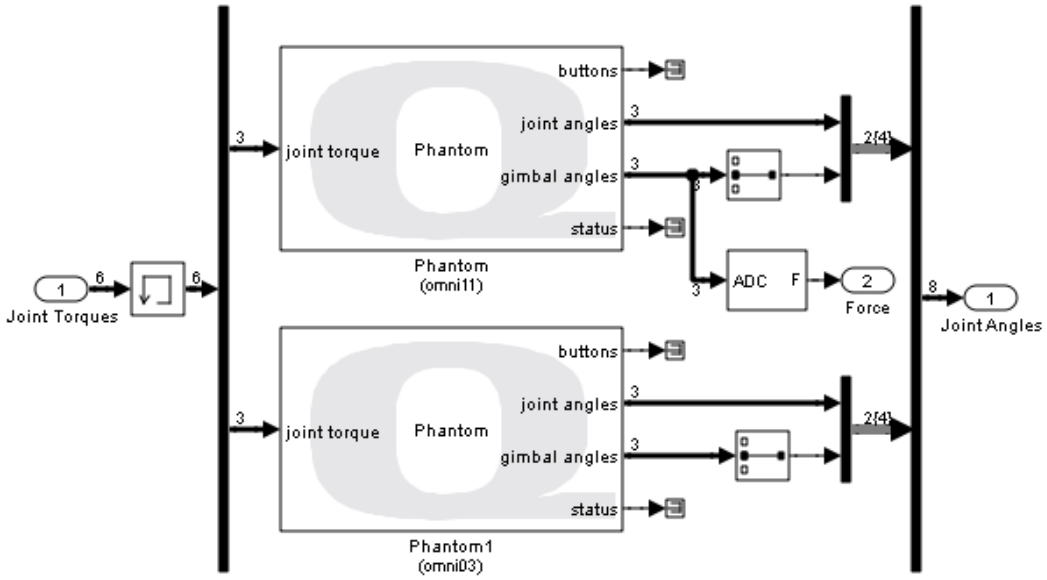

Figure 6.8 Subsystem for two Robots in the main model in Figure 6.7

In Figure 6.6, the inverse dynamics model is used to generate torques with given motion and force demands in advance of the main experiments. Note there are two point-to-point motion generators for up and down motions.

In Figure 6.7, the main Simulink model is shown, consisting of demand, controllers, outputs, and data logging. Figure 6.8 shows the subsystem of the inputs and outputs of Robots 1 and 2.

Angles and force are then recorded and analysed. All motions are executed for 11 cycles. Data from the first cycle is discarded to avoid any transient effects there may be when the robots start to move from their resting positions to initial positions. All time series plots show only one of the cycles for clarity. The last cycle is chosen, arbitrarily, to be displayed. The distributions of load angle errors and internal force errors in different controllers are also plotted. The results are presented in the following section.

6.2 Experimental Results

Same as previous chapter, time series of torque input components, angle errors and forces for each case are shown in the following pages, as well as their error distributions. Numerical results are reported as well.

6.2.1 Case II-a: One Active Robot and One Static Robot

In this case, Robot 2 is holding its position, while Robot 1 performs the motions defined in terms of load angle.

6.2.1.1 Time Series and Error Distributions

Table 6.5 Table of figures presented in Case II-a

	Slow point-to-point	**Fast point-to-point**	**SPHS**
Angle demand and errors	Figure 6.9	Figure 6.14	Figure 6.19
Angle error distributions	Figure 6.10	Figure 6.15	Figure 6.20
Demand and actual force	Figure 6.11	Figure 6.16	Figure 6.21
Force error distributions	Figure 6.12	Figure 6.17	Figure 6.22
Joint torque components	Figure 6.13	Figure 6.18	Figure 6.23

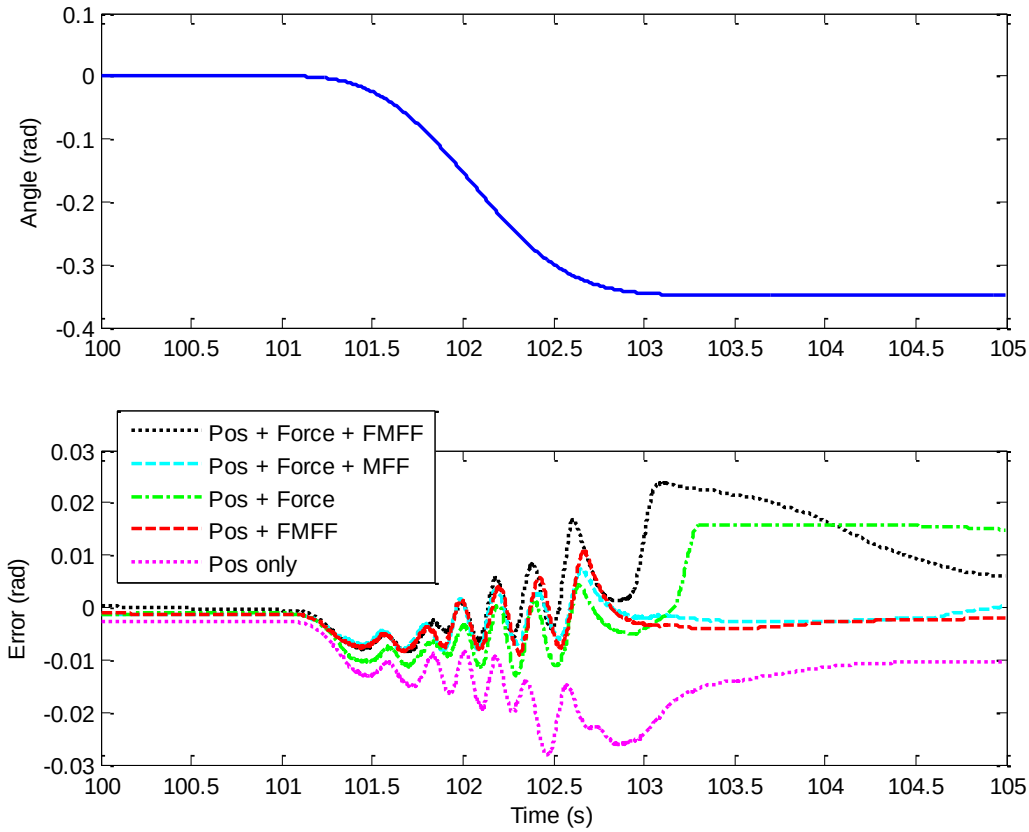

Figure 6.9 Case II-a experiment results: load angle in slow point-to-point motion. Top: angle demand. Bottom: angle errors in different controllers.

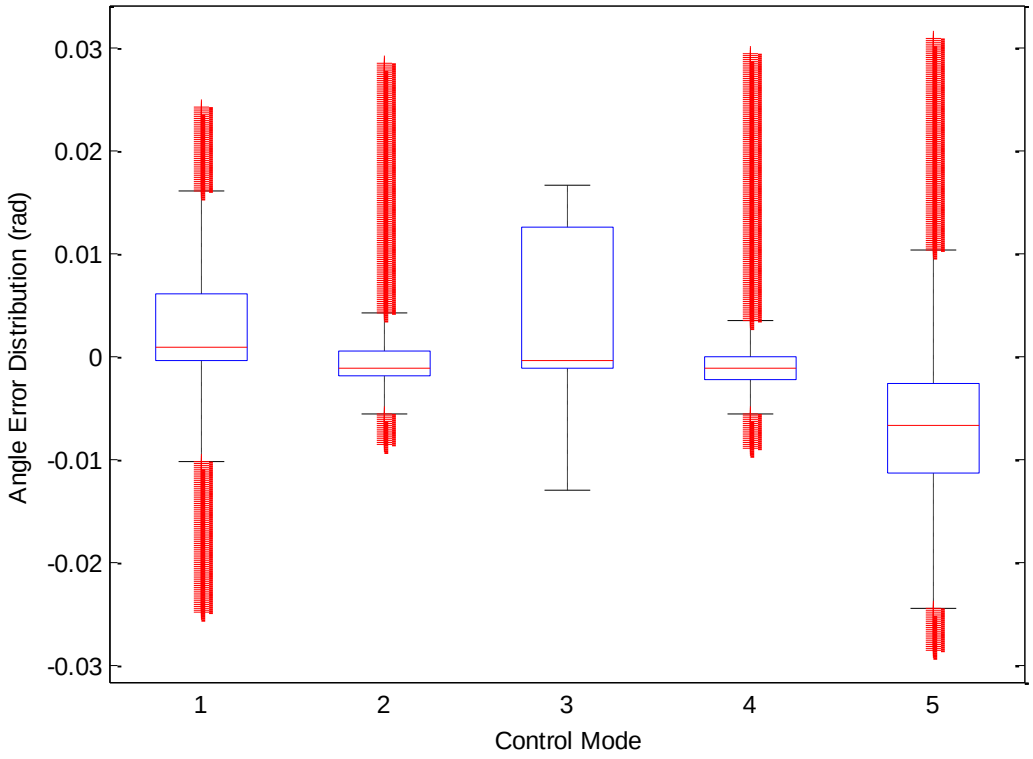

Figure 6.10 Case II-a experiment results: load angle error distribution in slow point-to-point motion.

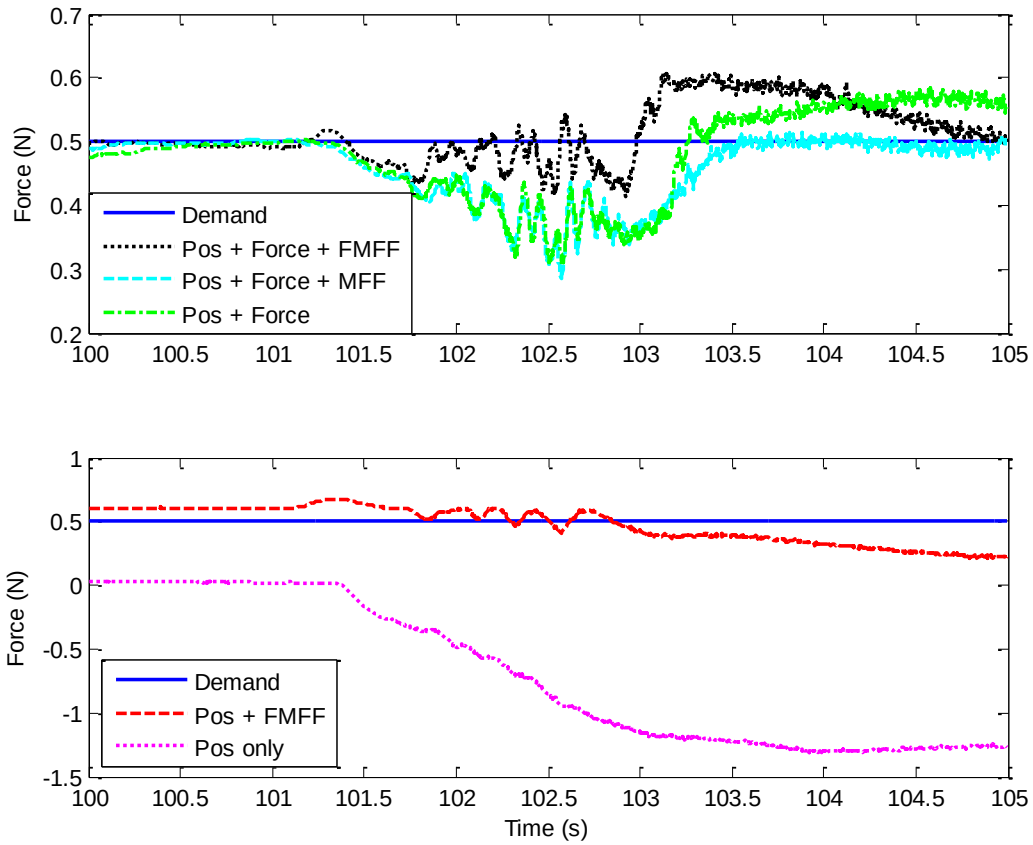

Figure 6.11 Case II-a experiment results: internal force in slow point-to-point motion. Top: controllers with force feedback. Bottom: controllers without force feedback.

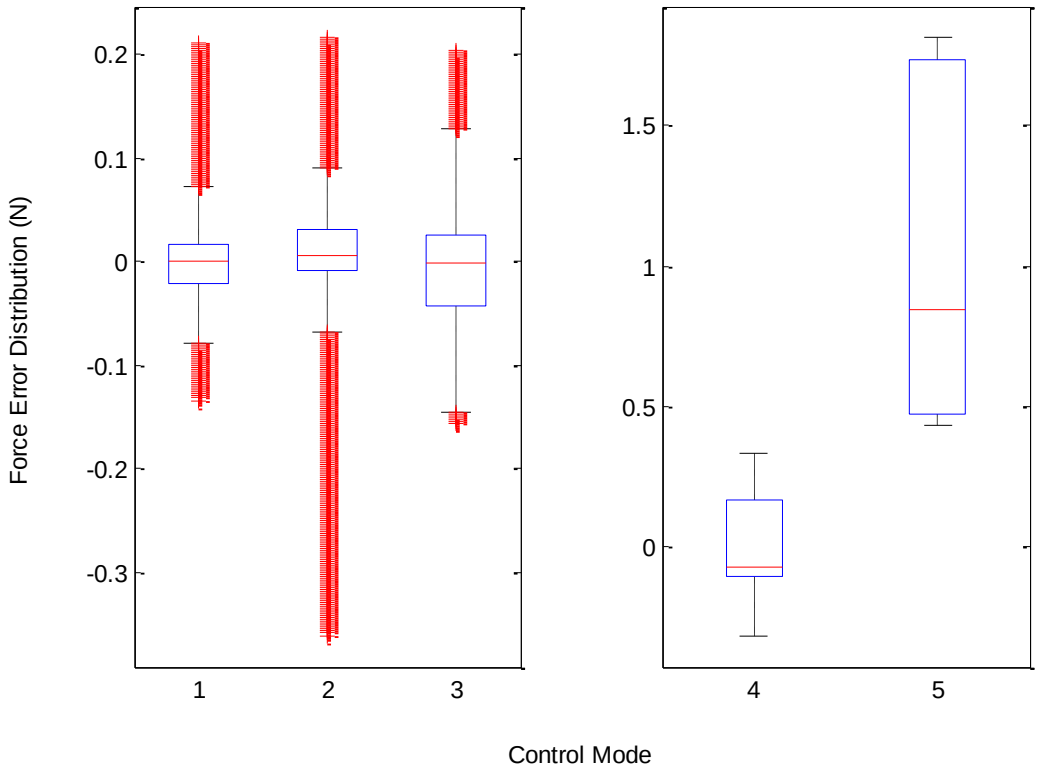

Figure 6.12 Case II-a experiment results: internal force error distribution in slow point-to-point motion.

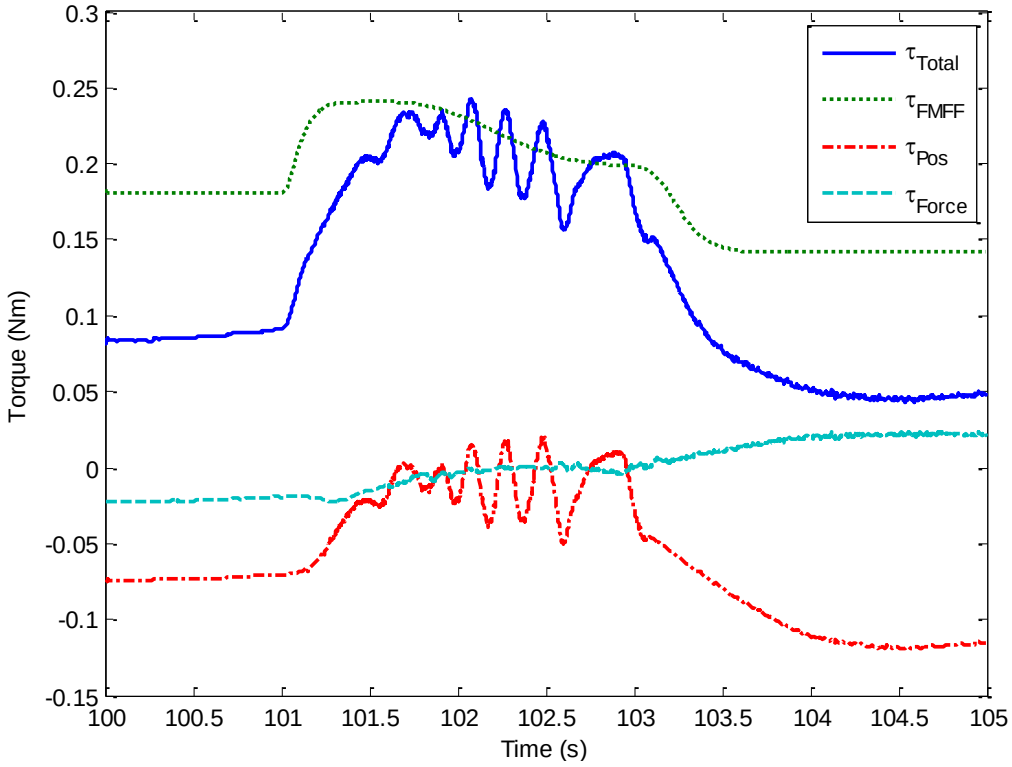

Figure 6.13 Case II-a experiment results: Controller 1 Joint 1 torque inputs in slow point-to-point motion.

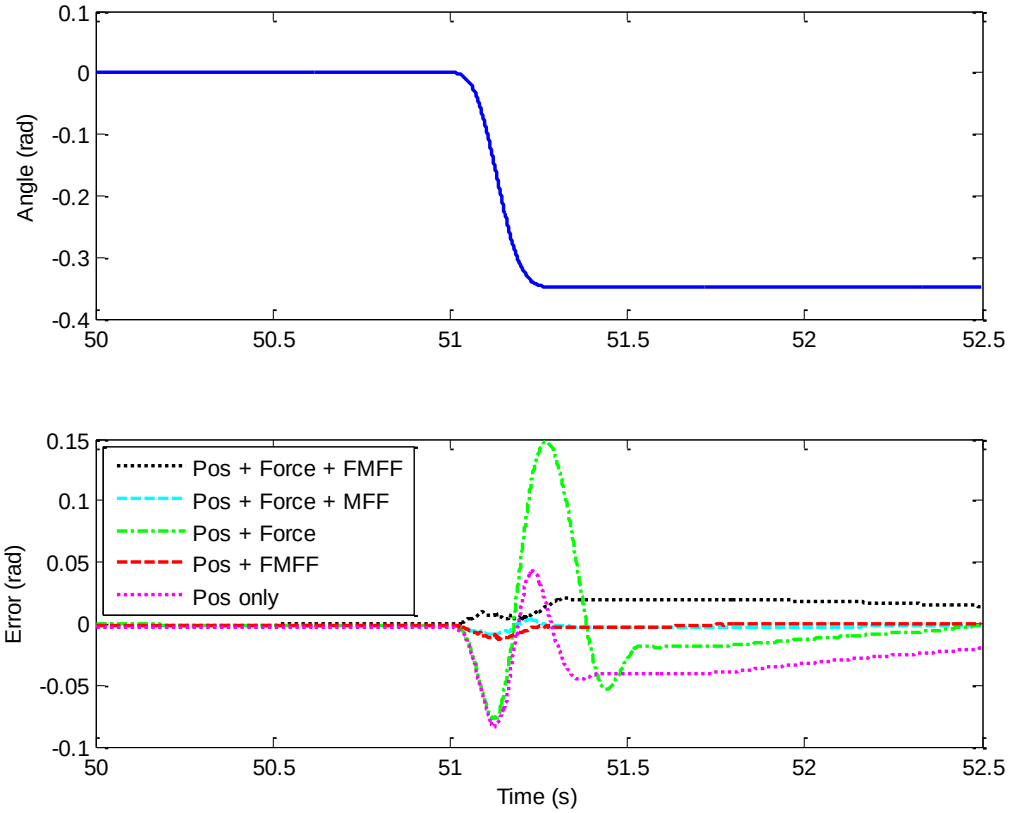

Figure 6.14 Case II-a experiment results: load angle in fast point-to-point motion. Top: angle demand. Bottom: angle errors of different controllers.

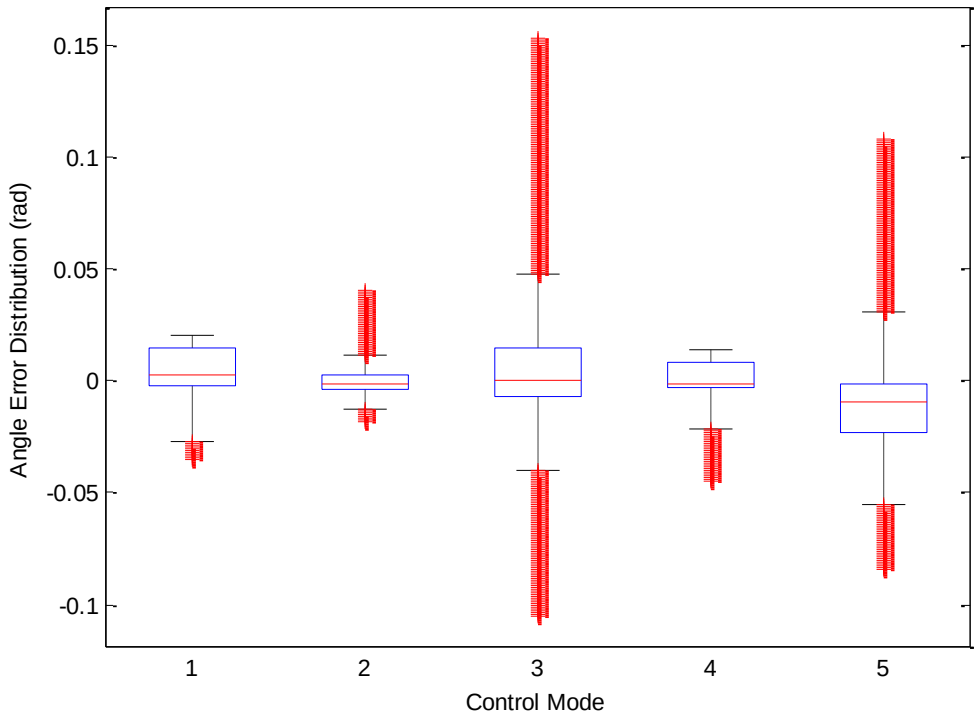

Figure 6.15 Case II-a experiment results: load angle error distribution in fast point-to-point motion.

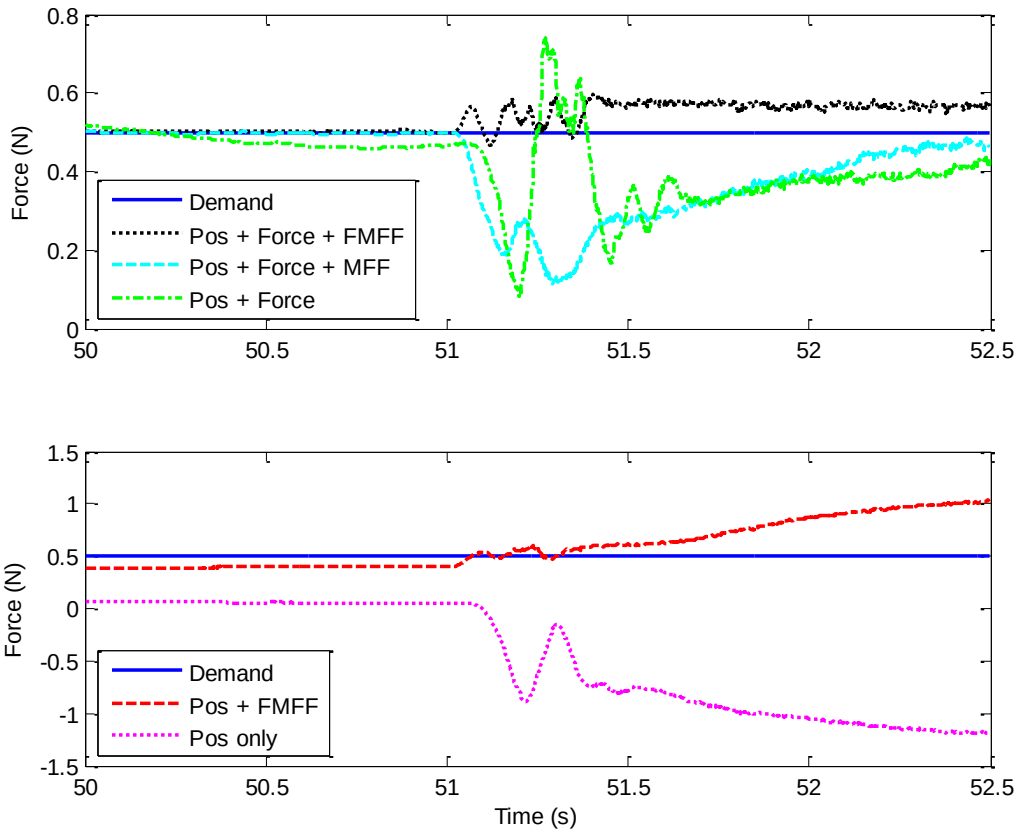

Figure 6.16 Case II-a experiment results: internal force in fast point-to-point motion. Top: controllers with force feedback. Bottom: controllers without force feedback.

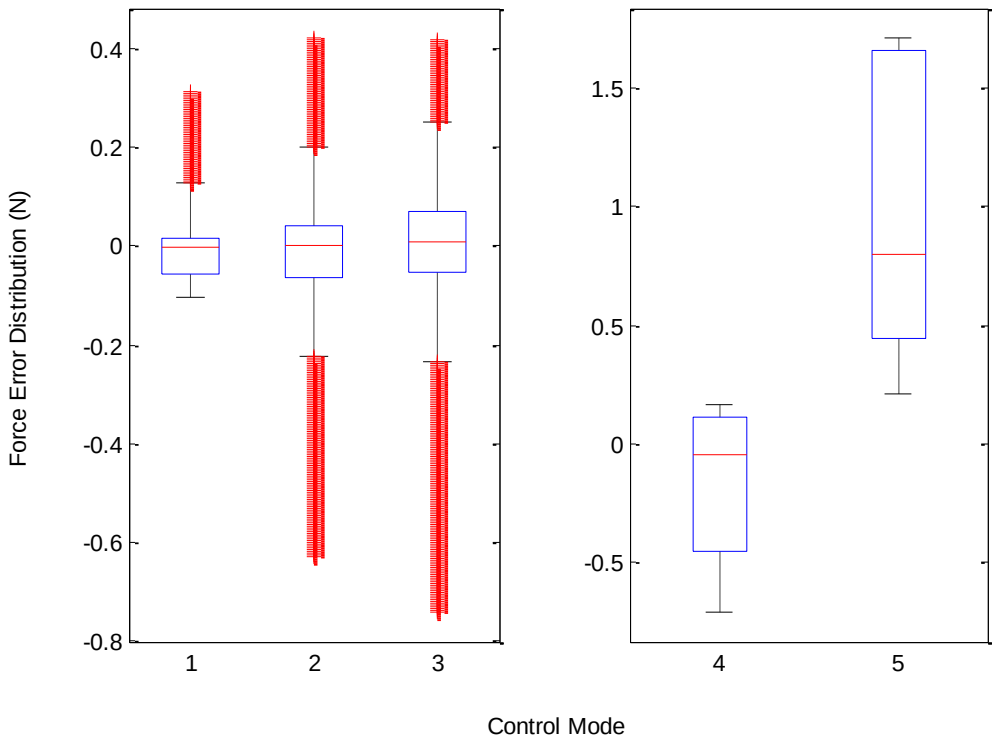

Figure 6.17 Case II-a experiment results: internal force error distribution in fast point-to-point motion. 1.Pos + Force + FMFF; 2.Pos + Force + MFF; 3.Pos + Force; 4.Pos + FMFF; 5.Pos only

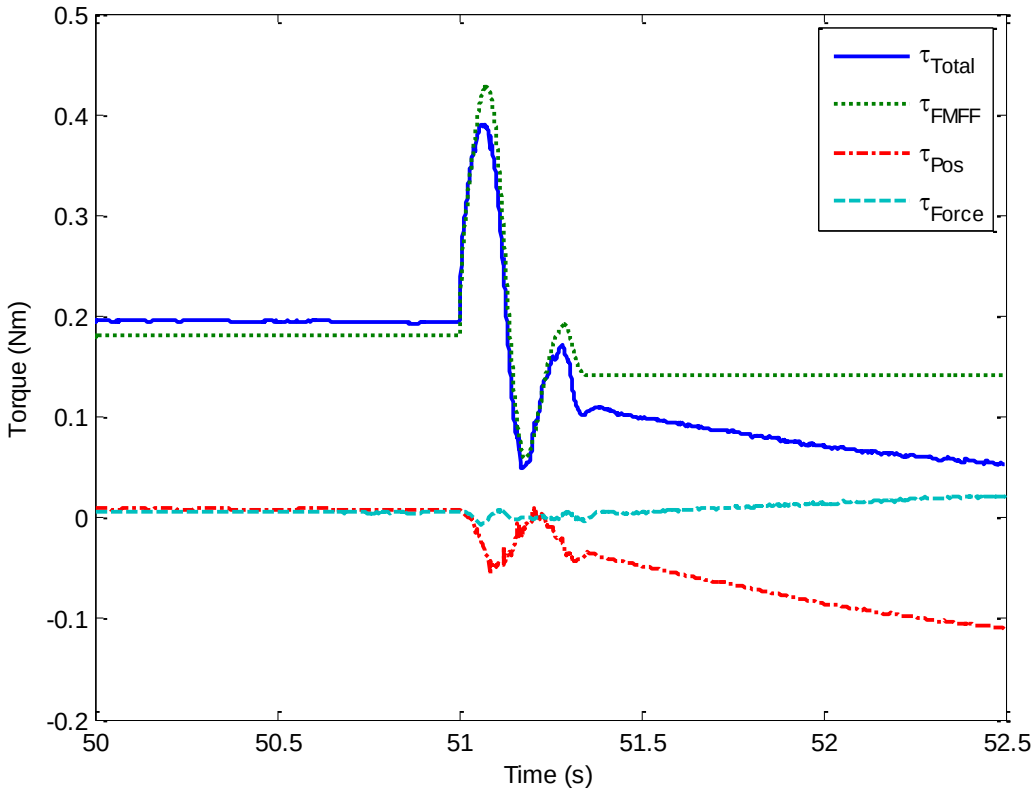

Figure 6.18 Case II-a experiment results: Controller 1 Joint 1 torque inputs in fast point-to-point motion.

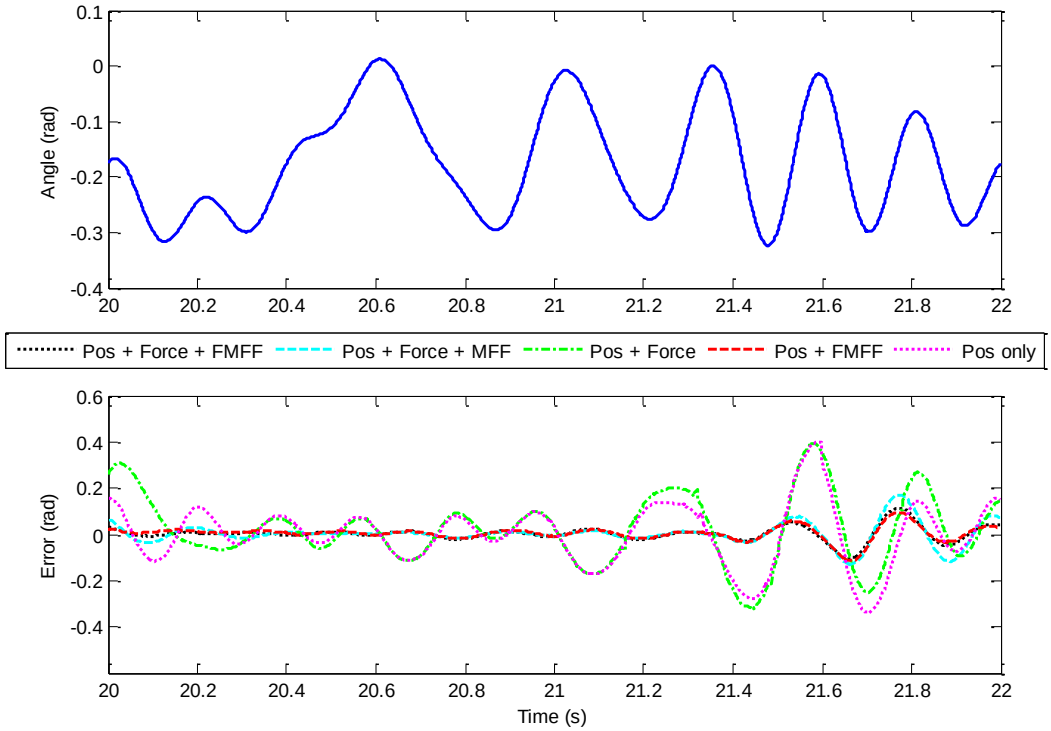

Figure 6.19 Case II-a experiment results: load angle in SPHS motion.
Top: angle demand. Bottom: angle errors of different controllers.

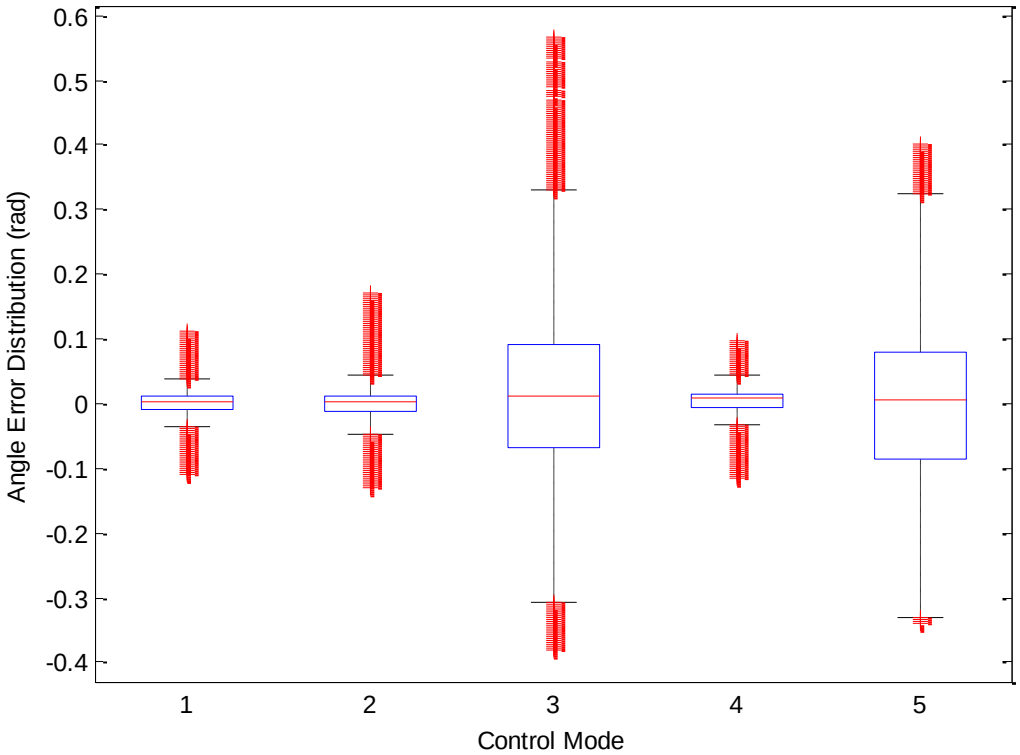

Figure 6.20 Case II-a experiment results: load angle error distribution in SPHS motion.
1.Pos + Force + FMFF; 2.Pos + Force + MFF; 3.Pos + Force; 4.Pos + FMFF; 5.Pos only

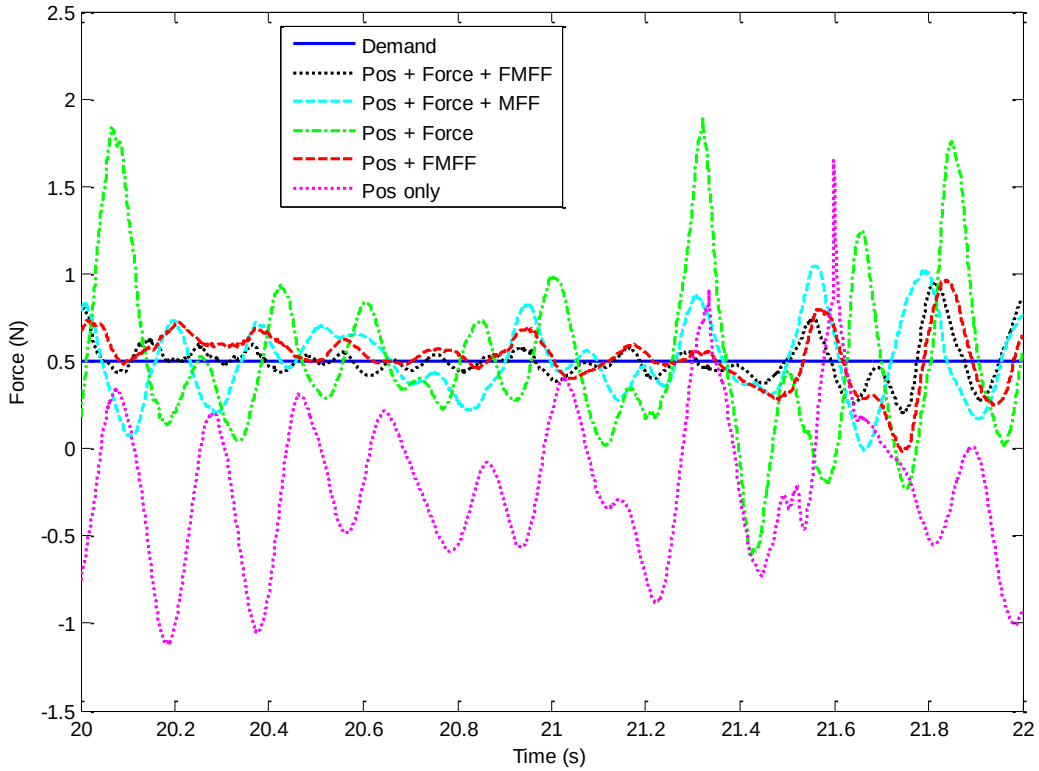

Figure 6.21 Case II-a experiment results: internal force in SPHS motion.

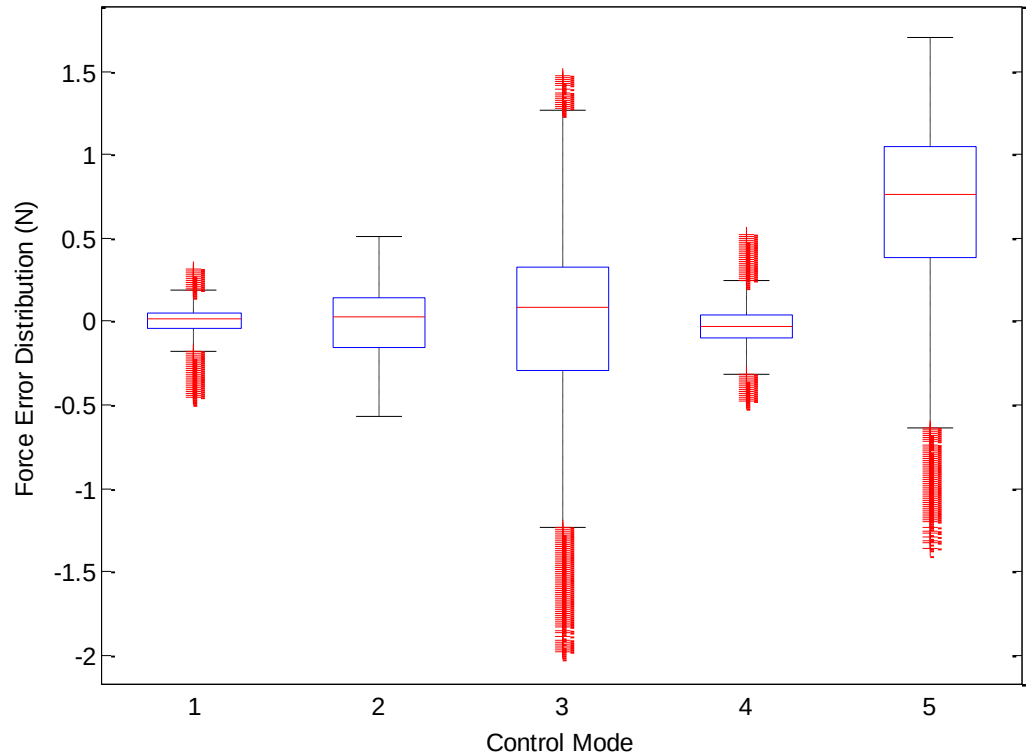

Figure 6.22 Case II-a experiment results: internal force error distribution in SPHS motion. 1.Pos + Force + FMFF; 2.Pos + Force + MFF; 3.Pos + Force; 4.Pos + FMFF; 5.Pos only

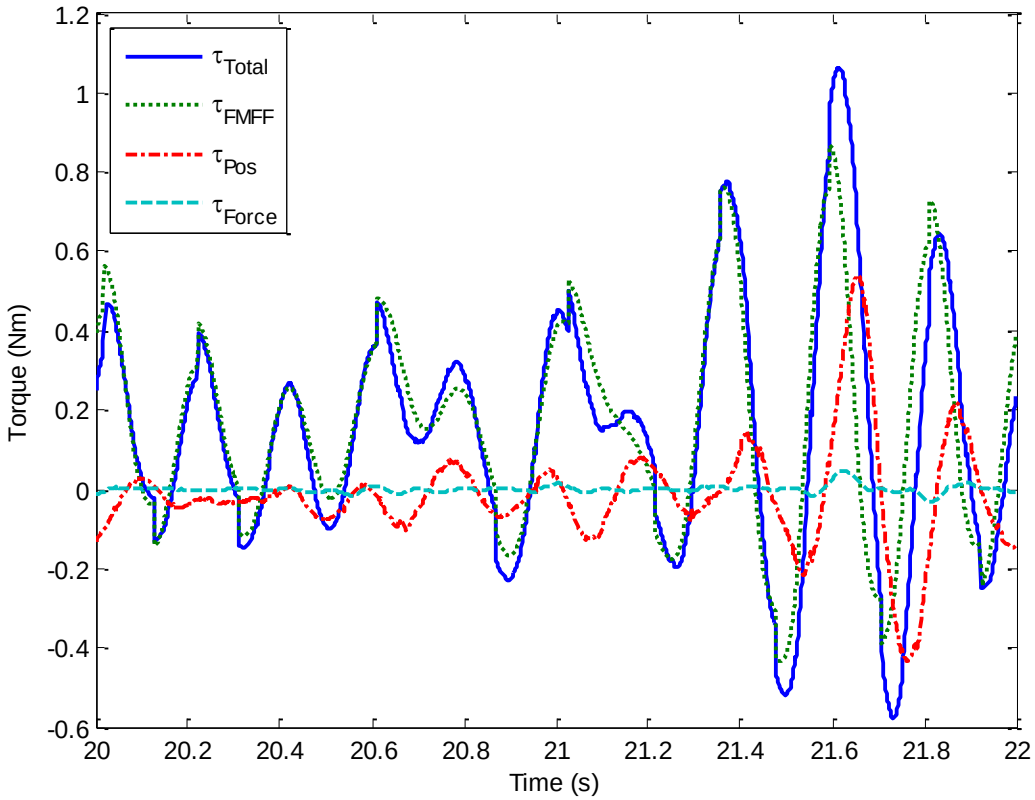

Figure 6.23 Case II-a experiment results: Controller 1 Joint 1 torque inputs in SPHS motion.

6.2.1.2 Numerical Results

The performances of the controllers are measured by RMSE of the joint angles. Table 6.6 to Table 6.8 list the position errors and force errors for different controllers from different motions. The error values in the tables are averaged over ten cycles for each motion. The highlighted cells in each column denote the smallest errors among different controllers.

Table 6.6 Position and force errors of *slow point-to-point* motion

Controller		Position RMSE (rad)	Force RMSE (N)
1	Pos + Force + FMFF	5.873×10^{-3}	3.604×10^{-2}
2	Pos + Force + MFF	4.540×10^{-3}	9.402×10^{-2}
3	Pos + Force	6.959×10^{-3}	9.453×10^{-2}
4	Pos + FMFF	5.251×10^{-3}	9.302×10^{-2}
5	Pos only	1.646×10^{-2}	1.065×10^{0}

Table 6.7 Position and force errors of *fast point-to-point* motion

Controller		Position RMSE (rad)	Force RMSE (N)
1	Pos + Force + FMFF	6.677×10^{-3}	3.842×10^{-2}
2	Pos + Force + MFF	9.266×10^{-3}	2.010×10^{-1}
3	Pos + Force	6.254×10^{-2}	1.897×10^{-1}
4	Pos + FMFF	4.648×10^{-3}	5.716×10^{-2}
5	Pos only	4.647×10^{-2}	8.655×10^{-1}

Table 6.8 Position and force errors of *SPHS* motion

Controller		Position RMSE (rad)	Force RMSE (N)
1	Pos + Force + FMFF	3.053×10^{-2}	1.116×10^{-1}
2	Pos + Force + MFF	4.523×10^{-2}	2.155×10^{-1}
3	Pos + Force	1.502×10^{-1}	5.196×10^{-1}
4	Pos + FMFF	2.983×10^{-2}	1.579×10^{-1}
5	Pos only	1.297×10^{-1}	8.582×10^{-1}

6.2.2 Case II-b: Two Active Robots

In this case, Robots 1 and 2 are working together to move the load while maintaining a constant compression force.

6.2.2.1 Time Series and Error Distributions

Table 6.9 Table of figures presented in Case II-b

	Slow point-to-point	Fast point-to-point	SPHS
Angle demand and errors	Figure 6.24	Figure 6.29	Figure 6.34
Angle error distributions	Figure 6.25	Figure 6.30	Figure 6.35
Demand and actual force	Figure 6.26	Figure 6.31	Figure 6.36
Force error distributions	Figure 6.27	Figure 6.32	Figure 6.37
Joint torque components	Figure 6.28	Figure 6.33	Figure 6.38

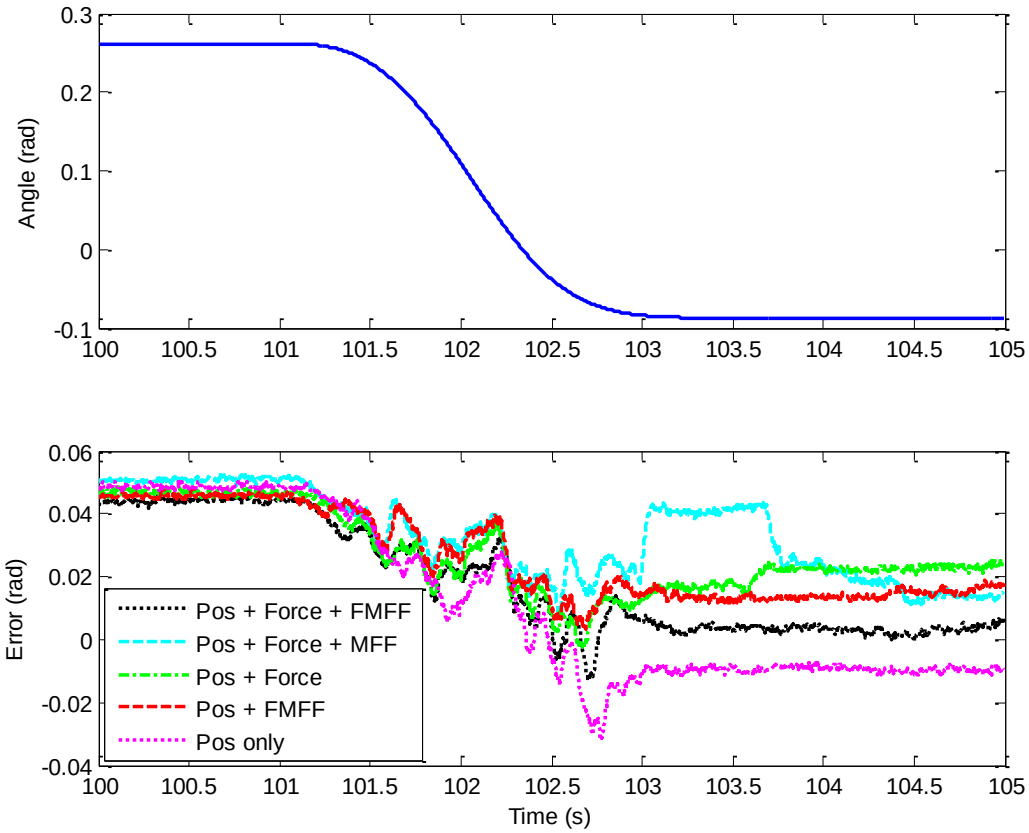

Figure 6.24 Case II-b experiment results: load angle in slow point-to-point motion. Top: angle demand. Bottom: angle errors in different controllers.

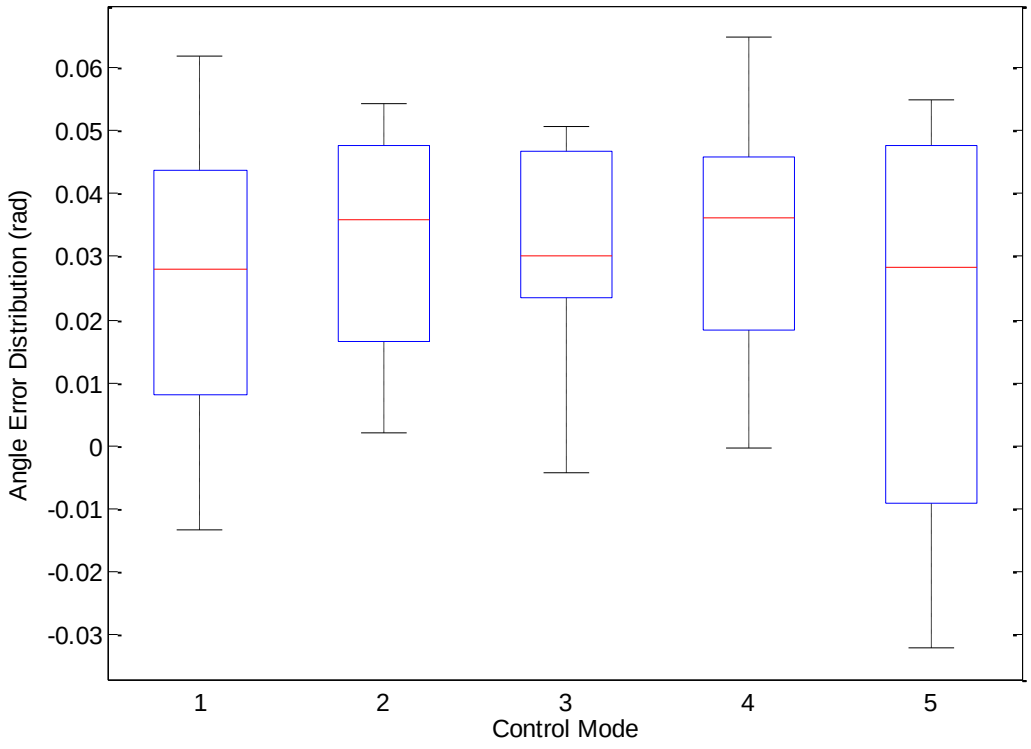

Figure 6.25 Case II-b experiment results: load angle error distribution in slow point-to-point motion.

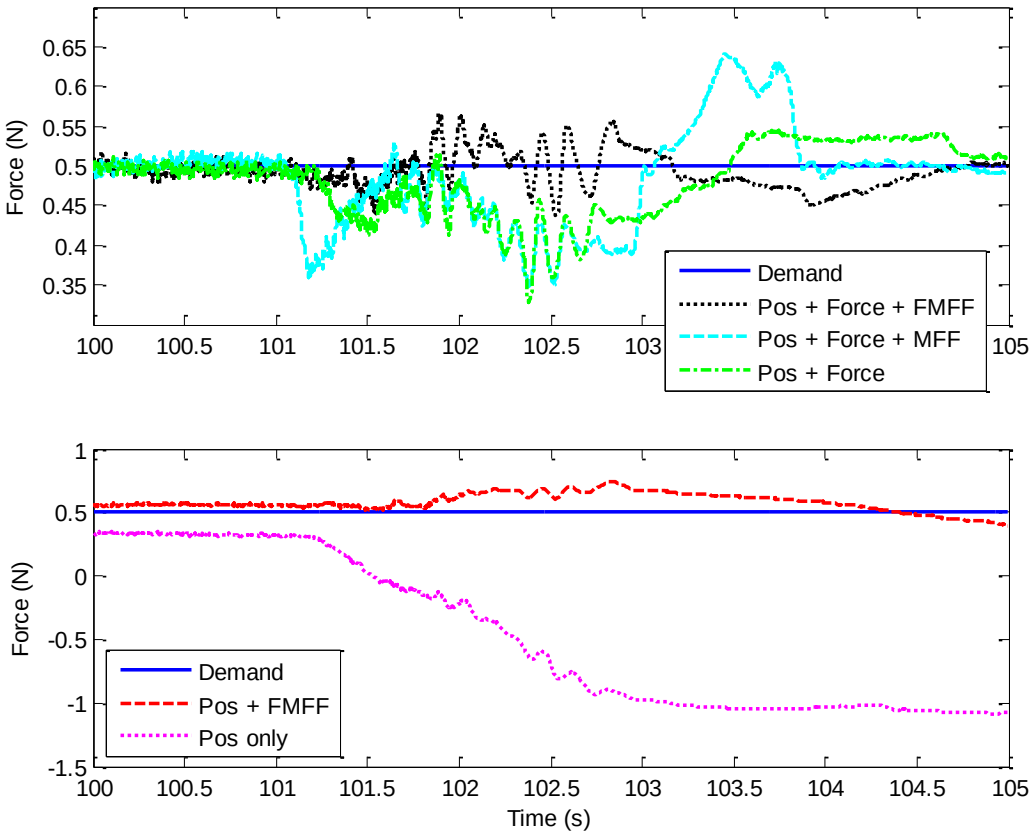

Figure 6.26 Case II-b experiment results: internal force in slow point-to-point motion. Top: controllers with force feedback. Bottom: controllers without force feedback.

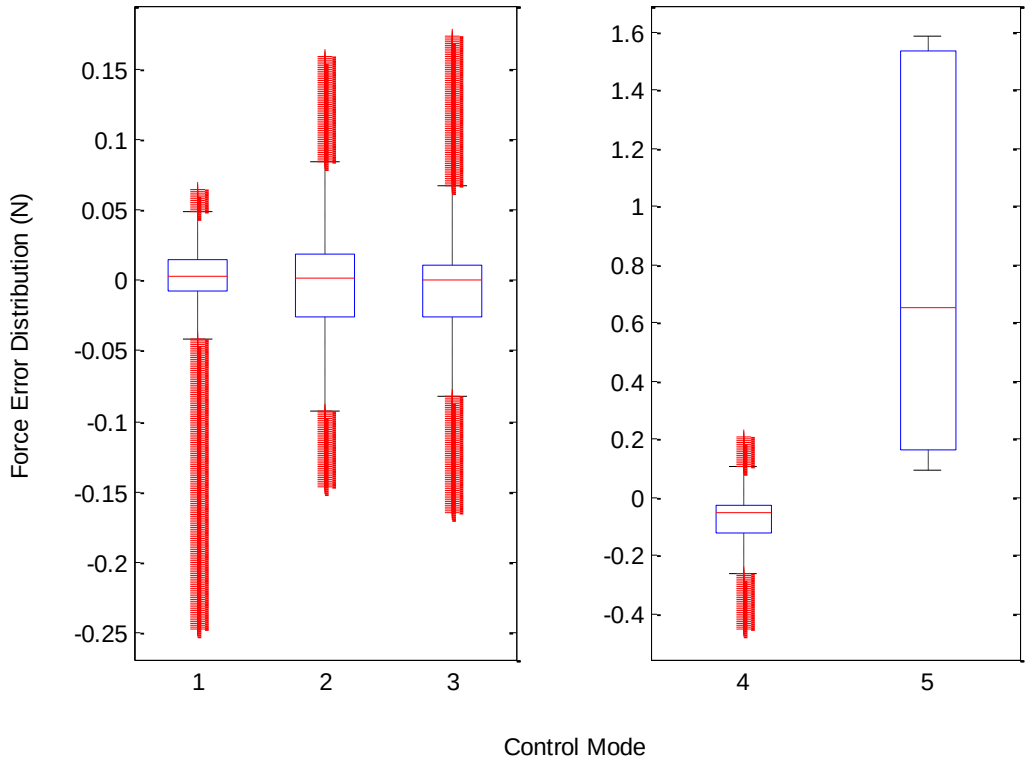

Figure 6.27 Case II-b experiment results: internal force error distribution in slow point-to-point motion.

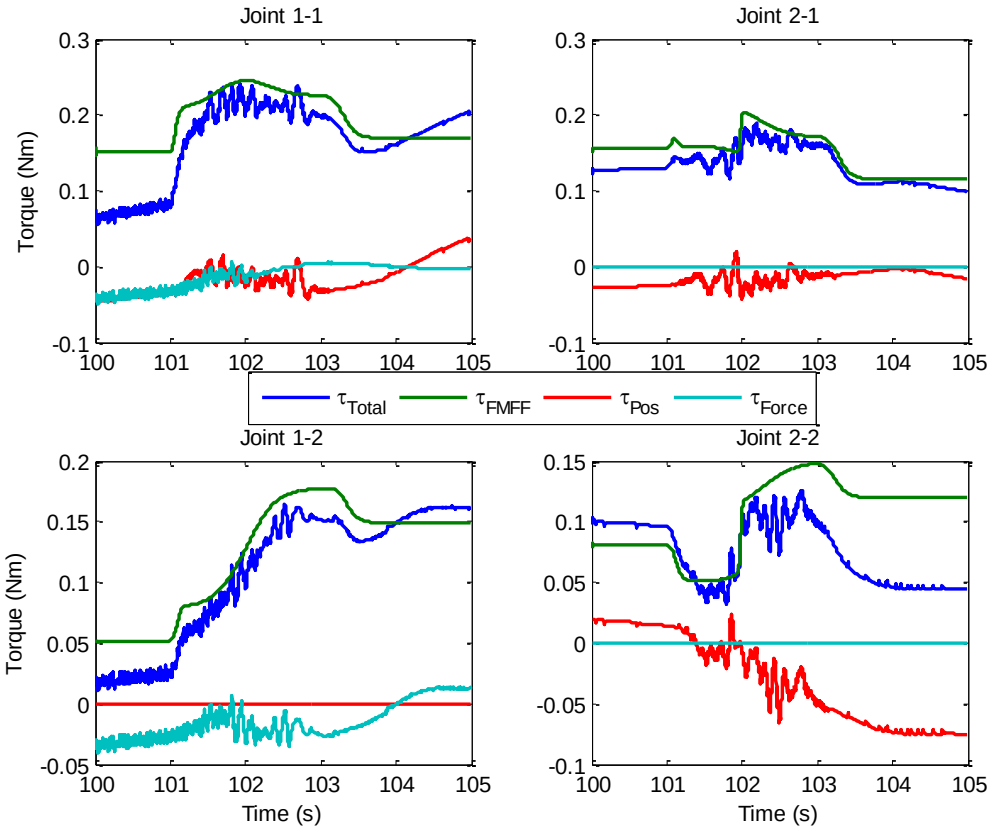

Figure 6.28 Case II-b experiment results: Controller 1 torque inputs in slow point-to-point motion.

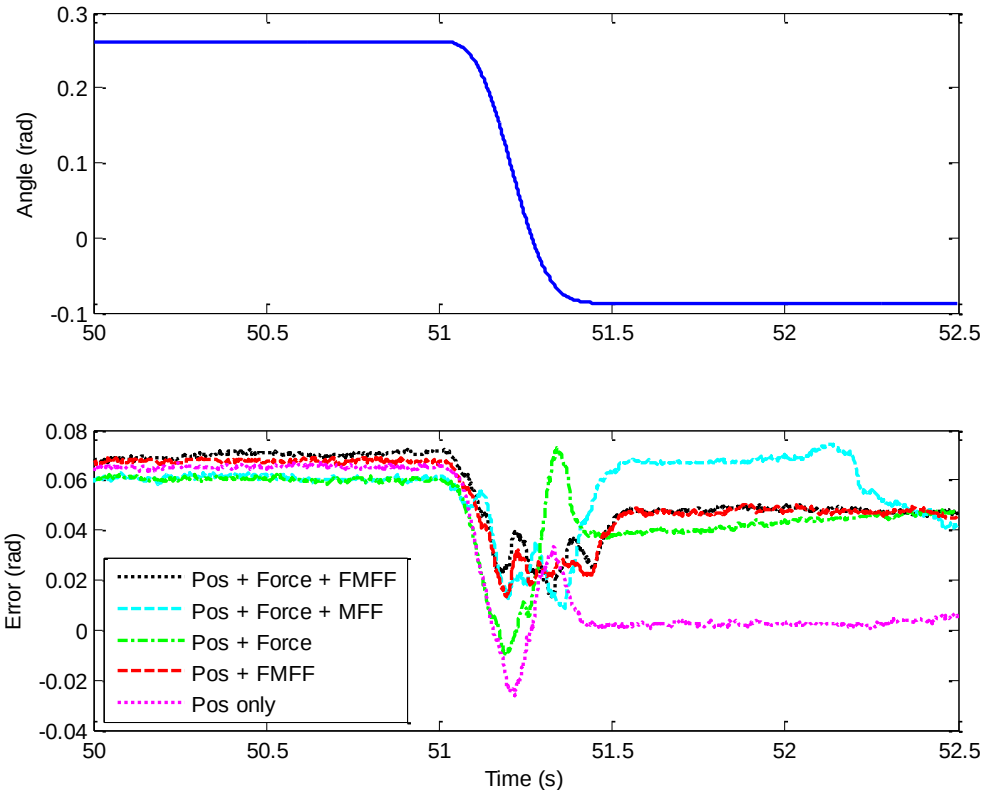

Figure 6.29 Case II-b experiment results: load angle in fast point-to-point motion. Top: angle demand. Bottom: angle errors of different controllers.

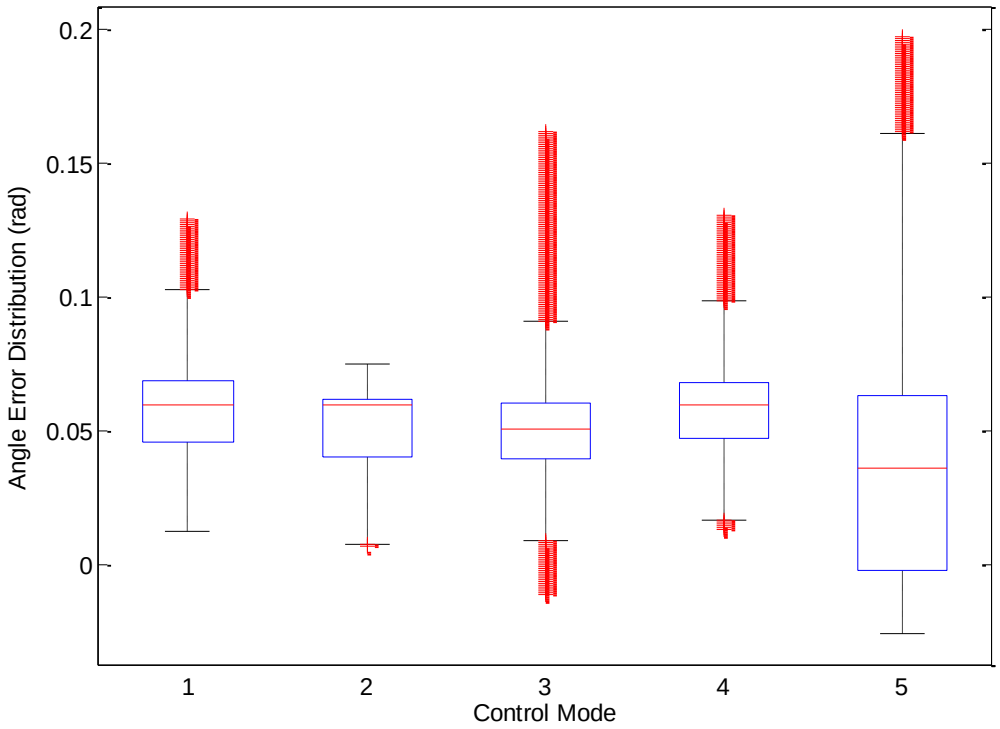

Figure 6.30 Case II-b experiment results: load angle error distribution in fast point-to-point motion.

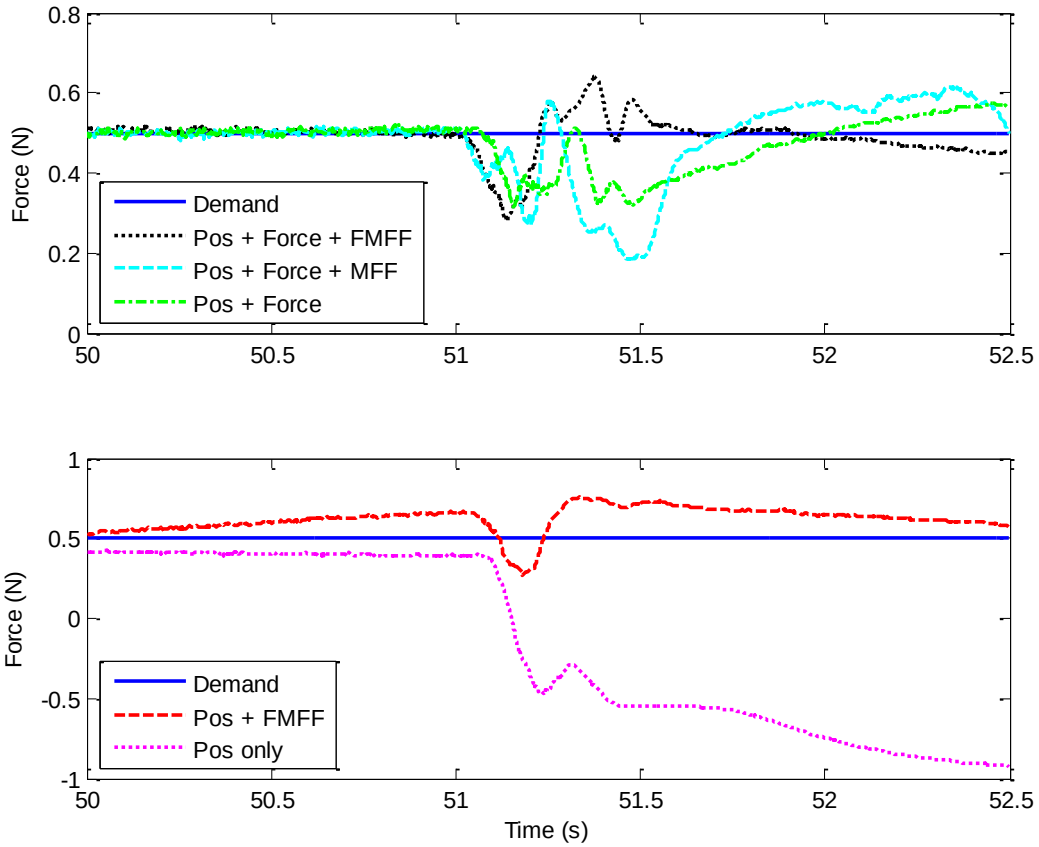

Figure 6.31 Case II-b experiment results: internal force in fast point-to-point motion. Top: controllers with force feedback. Bottom: controllers without force feedback.

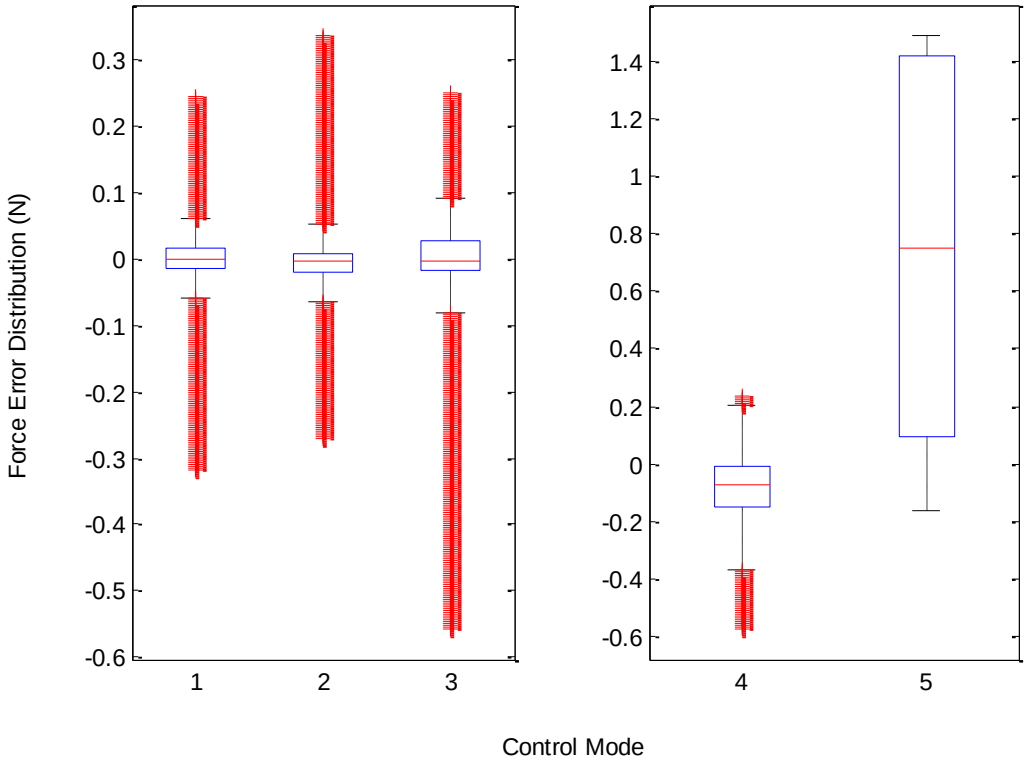

Figure 6.32 Case II-b experiment results: internal force error distribution in fast point-to-point motion. 1.Pos + Force + FMFF; 2.Pos + Force + MFF; 3.Pos + Force; 4.Pos + FMFF; 5.Pos only

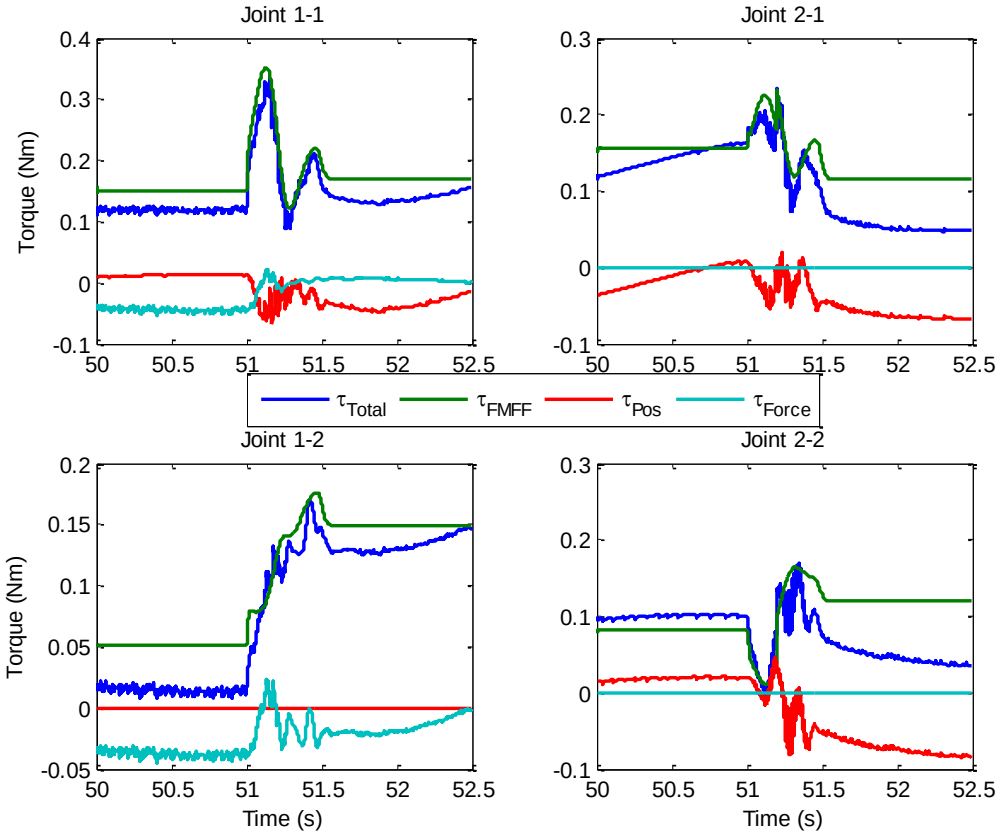

Figure 6.33 Case II-b experiment results: Controller 1 torque inputs in fast point-to-point motion.

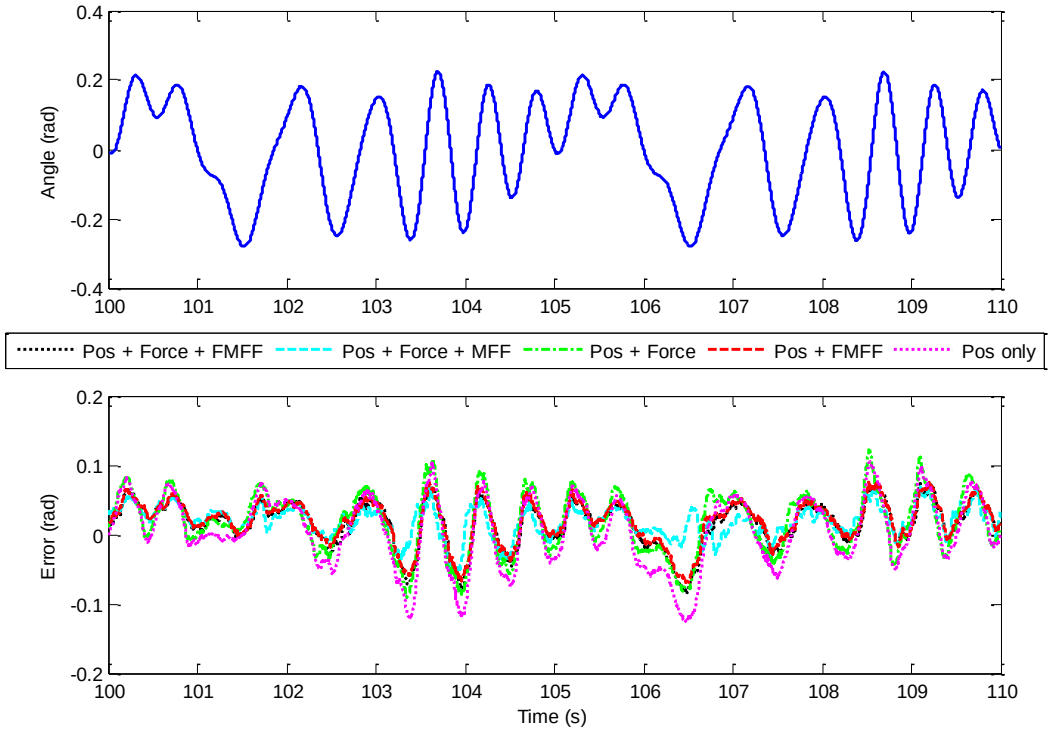

Figure 6.34 Case II-b experiment results: load angle in SPHS motion.
Top: angle demand. Bottom: angle errors of different controllers.

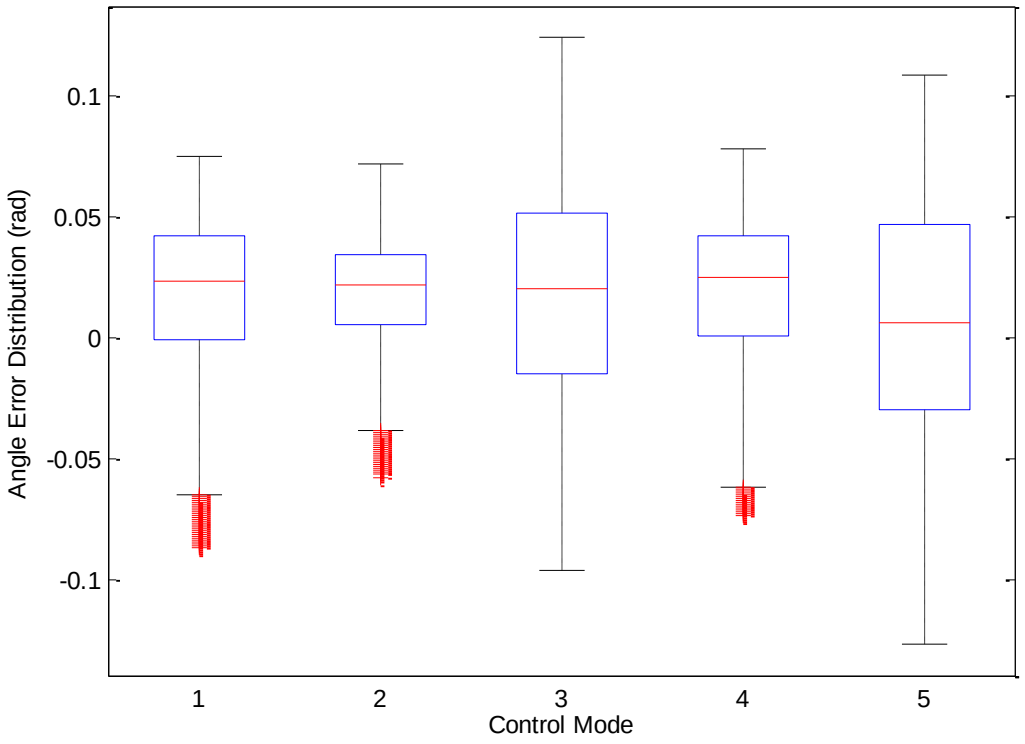

Figure 6.35 Case II-b experiment results: load angle error distribution in SPHS motion.
1.Pos + Force + FMFF; 2.Pos + Force + MFF; 3.Pos + Force; 4.Pos + FMFF; 5.Pos only

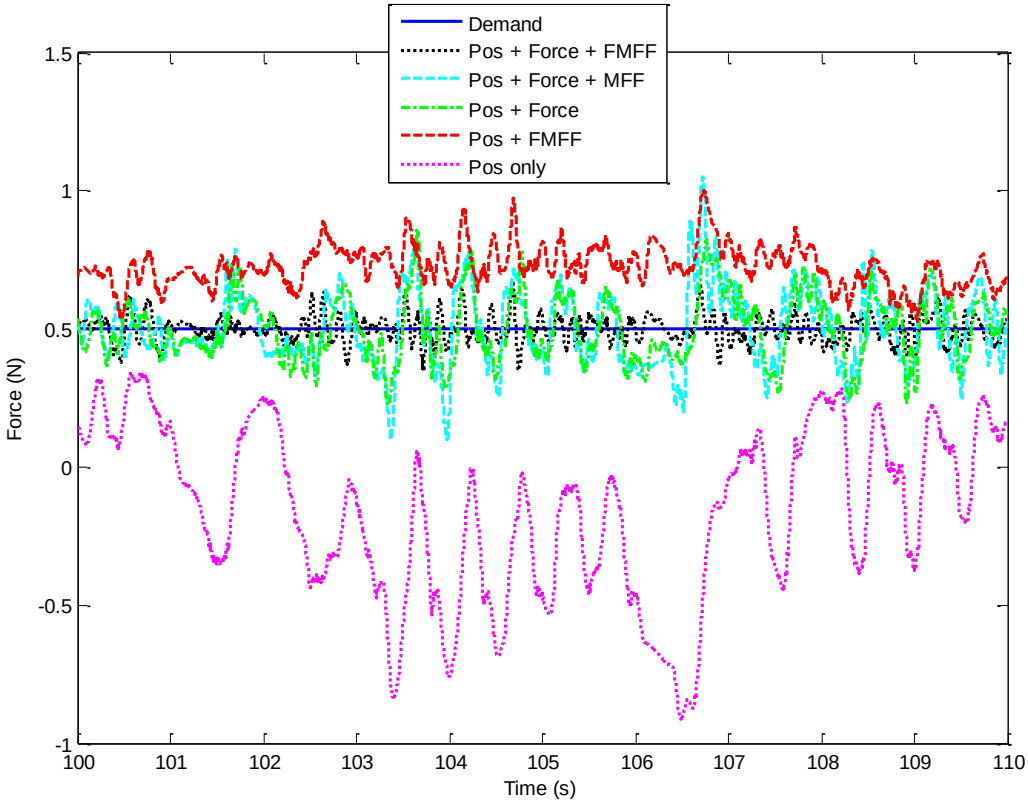

Figure 6.36 Case II-b experiment results: internal force in SPHS motion.

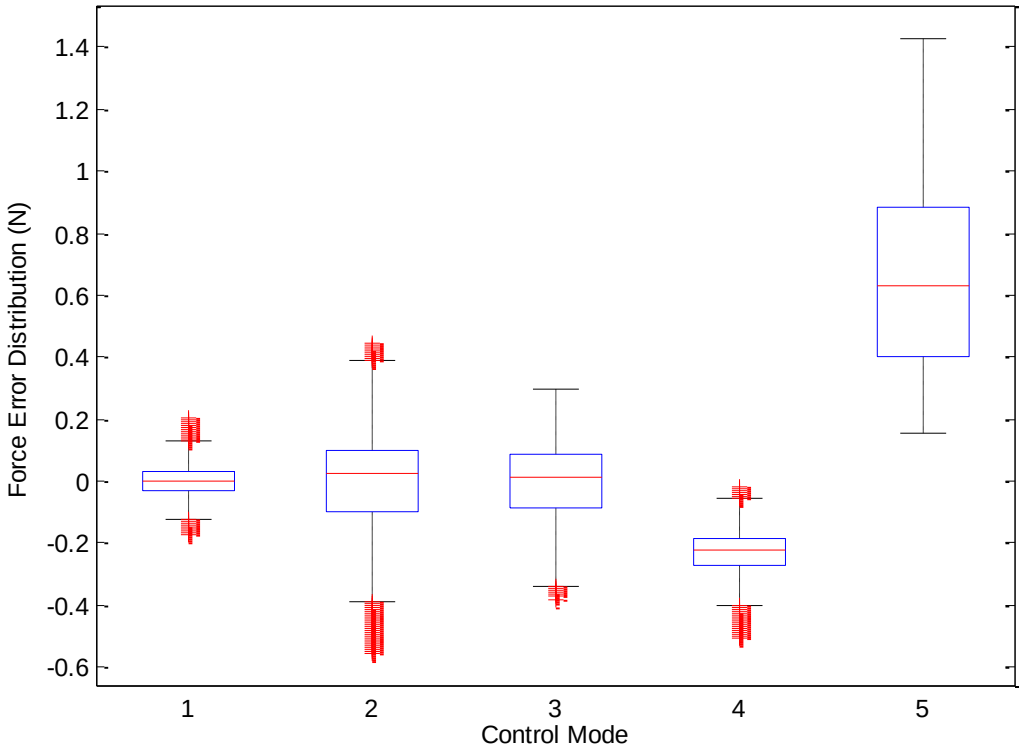

Figure 6.37 Case II-b experiment results: internal force error distribution in SPHS motion. 1.Pos + Force + FMFF; 2.Pos + Force + MFF; 3.Pos + Force; 4.Pos + FMFF; 5.Pos only

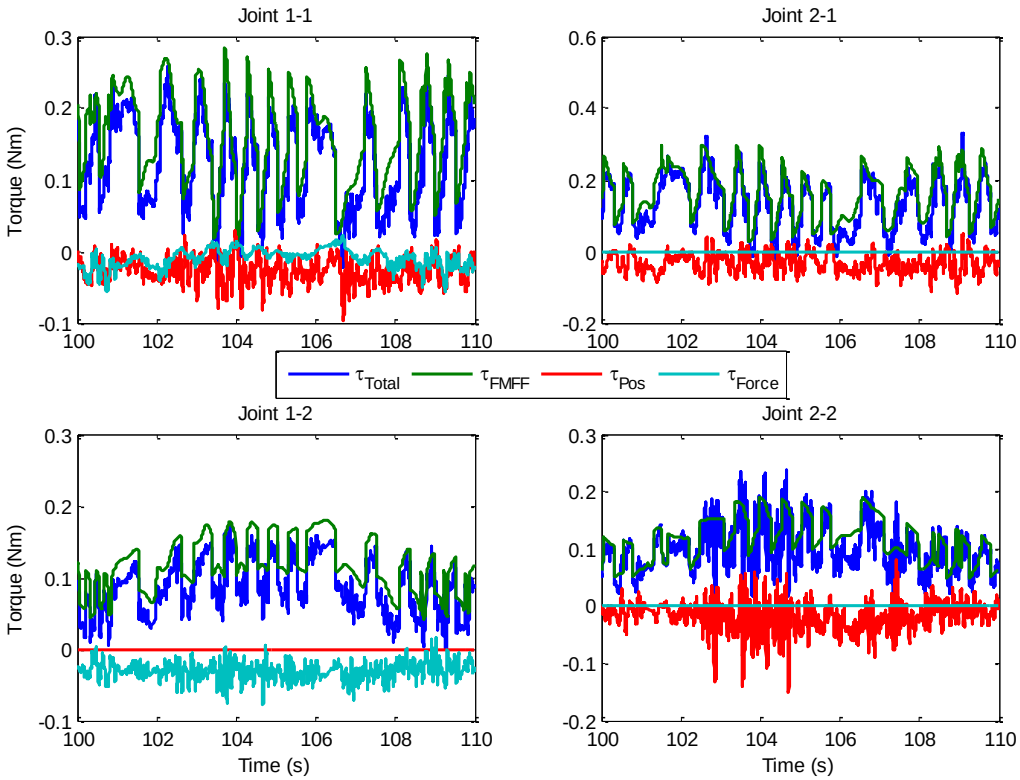

Figure 6.38 Case II-b experiment results: Controller 1 torque inputs in SPHS motion.

6.2.2.2 Numerical Results

Table 6.10 to Table 6.12 list the position errors and force errors for different controllers from different motions. The error values in the tables are averaged over ten cycles for each motion. The highlighted cells denote the smallest errors among different controllers.

Table 6.10 Position and force errors of *slow point-to-point* motion

Controller		Position RMSE (rad)	Force RMSE (N)
1	Pos + Force + FMFF	1.460×10^{-2}	2.827×10^{-2}
2	Pos + Force + MFF	2.479×10^{-2}	8.017×10^{-2}
3	Pos + Force	1.827×10^{-2}	6.712×10^{-2}
4	Pos + FMFF	1.973×10^{-2}	1.311×10^{-1}
5	Pos only	1.473×10^{-2}	9.036×10^{-1}

Table 6.11 Position and force errors of *fast point-to-point* motion

Controller		Position RMSE (rad)	Force RMSE (N)
1	Pos + Force + FMFF	1.100×10^{-2}	1.096×10^{-1}
2	Pos + Force + MFF	2.135×10^{-2}	1.395×10^{-1}
3	Pos + Force	3.183×10^{-2}	1.022×10^{-1}
4	Pos + FMFF	9.678×10^{-3}	1.791×10^{-1}
5	Pos only	3.084×10^{-2}	6.869×10^{-1}

Table 6.12 Position and force errors of *SPHS* motion

Controller		Position RMSE (rad)	Force RMSE (N)
1	Pos + Force + FMFF	3.740×10^{-2}	4.853×10^{-2}
2	Pos + Force + MFF	3.696×10^{-2}	1.388×10^{-1}
3	Pos + Force	4.119×10^{-2}	1.173×10^{-1}
4	Pos + FMFF	3.601×10^{-2}	2.402×10^{-1}
5	Pos only	5.381×10^{-2}	7.303×10^{-1}

6.3 Analysis and Comparison

In general, as in Case I, the faster the motion, the better performance the feedforward controllers deliver when compared with the other controllers. Amongst all motions, the best force tracking is always achieved by Controller 1 except for once where Controller 3 is slightly better; however, with much poorer position tracking.

6.3.1 Case II-a: One Active Robot and One Static Robot

Similar to Case I, Case II-a has an almost identical pattern for the performances. Controllers 1, 2 and 4 are better at position tracking and Controllers 1 and 4 are better at force tracking. This means the extended inverse dynamics controller always improves the performance of both position and force tracking. It only affects the performance of position tracking slightly in slower speed motion. This may be caused by the modelling errors of the joint frictions.

Figure 6.13, Figure 6.18, and Figure 6.23 reveal the torque inputs of Joint 1 and the contributions from different control loops. They all show that the feedforward loop makes up the most of the input signals, which means the operational and practical effectiveness of the controller is demonstrated.

6.3.2 Case II-b: Two Active Robots

Similar to the previous cases, controllers with the force feedforward loop (Controllers 1 and 4) show very promising performance. Again, take Controller 3's performance as a benchmark. In slow point-to-point motion, Controller 1 improves both position and force tracking. In contrast, Controller 2 worsens both. In fast point-to-point and SPHS motions, both Controllers 1 and 2 improve position tracking, but only Controller 1 improves force tracking.

When comparing Controllers 1 and 2, in point-to-point motions, the former performs better for both position and force tracking; and in SPHS motion, it performs better for force tracking and similar for position tracking.

Figure 6.28, Figure 6.33, and Figure 6.38 reveal the torque inputs of all four joints and the contributions from different control loops. They all show that the feedforward loop makes up the most of the input signals.

6.4 Concluding Remarks

This chapter gives the details of Case II experimental set-ups. New motions are designed for the experiments. The results are then reported and analysed from the experiments on Case II-a and Case II-b. Results show that Controller 1, which is based on the extended inverse dynamics controller, has the best combined position and force tracking.

7 Conclusions and Future Work

7.1 Conclusions

This thesis develops a novel model-based method to control the position and interaction forces during a cooperative robot handling task. Although the method is very generic and can be applied to many different systems, the thesis uses a task of moving a shared object using two robots as a case study. The method developed is based on using an accurate dynamic model of the robots and the object being handled, and computing the forces required to produce the motion as well as force resulting from the constraints. In order to produce accurate models of the robots and manipulated objects, the thesis develops a novel approach for designing optimal excitation trajectories for dynamic parameter estimation. This new technique improves the quality of the parameters inferred from the data thus improves the fidelity of the models. This improvement on the parameters also means an improvement in the control performance of the cooperative system.

This thesis makes two novel contributions to the field of robotics. First, an extended definition of force and motion in inverse dynamics analysis allows the calculation of internal force to be performed with conventional inverse dynamics analysis. Second, the use of SPHS signals simplifies the process of finding optimal excitation trajectories for manipulator dynamic parameter estimation.

Conventional inverse dynamics models only take motion demand as inputs to calculate required control signals. Internal and external forces are calculated separately through inverse kinematics then added to the control inputs. The proposed extended inverse dynamic formulation integrates both force and motion inputs in predefined coordinates, which simplifies calculation in control applications. The inverse dynamics equations of the entire system have been derived based on the constrained Lagrangian formulation. The proposed control method takes both robots and load dynamics into consideration. The use of redundant coordinates allows different coordinates to be used to specify the trajectory, control inputs, and motion measurement.

The extended inverse dynamics model is used to control the motion of the robots as well as the internal force applied on the load. The effectiveness of the method is verified by simulation and experiments. The experiments utilise both low speed and high-speed motions. Results show that the proposed controller has better position and force tracking in higher speed motions, compared to traditional hybrid position/force controllers.

The new method of calculating feedforward control inputs could be beneficial to many industrial and research applications. The method simplifies the computation of inverse dynamic controllers of manipulators under the influence of known or modelled internal or external force or torque, compared to methods previously stated in the literature. It has great potential of improving the performance of online calculation. Furthermore, generalized method is derived using Lagrangian method, which is applicable to all types of manipulators.

In the new approach for finding optimal excitation trajectories, SPHS signals have been utilised so that the synchronisation of the signals can lead to maximum system information extraction. An SPHS signal has the advantage of having only three parameters to optimise, which leads to a method with a reduced search space when compared to traditional methods such as the Fourier series. The method also guarantees an optimal solution because it does not use generic algorithms where the optimisation may result in local minima.

The proposed method is tested experimentally on an Omni robot. The model estimated from the optimised trajectories shows a better joint torque prediction compared to the model estimated from non-optimised trajectories. The resulting models are also tested as part of a feedforward controller in a position-tracking task. The results show that the optimised model has more improvement over a PID-only controller in tracking the robot’s position compared with the non-optimised model.

In the field of robotic control, accurate dynamic models are essential to the performance of model-based controllers. The new designing method reduces the effort to obtain excitation trajectories for experimental parameter estimation.

7.2 Recommendations for Future Work

The following recommendations can be considered if further research is to be continued.

- The feedforward part of the controller is provided by running the inverse dynamics model offline prior to the main experiment. If the Dysim Simulink block can be made compatible with the Quanser QuaRC environment, then the feedforward model is able to be applied in real-time.
- Adaptive control law can be added to alter the parameters of the inverse dynamics model online in order to better handle the unmodelled robot and/or load dynamics.
- The force sensor currently used has only one axis. This has not been a problem in the experiments carried out during this research. However, a six-axis force/torque sensor is needed if the controller is to be expanded into three-dimensional space. The data acquisition of the force sensor is accomplished by one of the ADC channels inside the Omni Robot. An external arrangement that is compatible with the QuaRC environment is needed for the six-axis force/torque sensor.

References

1. Liu, J. and Wu, J., *Multi-agent robotic systems*. 2001, Boca Raton, Fla. ; London: CRC Press. xvii, 304 p.
2. Nof, S.Y. and SpringerLink (Online service), *Springer Handbook of Automation*. 2009, Springer Berlin Heidelberg: Berlin, Heidelberg.
3. Kurfess, T.R., *Robotics and automation handbook*. 2005, Boca Raton, Fla. ; London: CRC.
4. Colestock, H., *Industrial robotics : selection, design, and maintenance*. 2005, New York ; London: McGraw-Hill Professional. xi, 211 p.
5. Kröger, T. and Wahl, F. *Low-Level Control of Robot Manipulators: A Brief Survey on Sensor-Guided Control and On-Line Trajectory Generation*. in *Proceedings of the IEEE ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications–How to Modify and Enhance Commercial Controllers, Anchorage*. 2010.
6. Dai, J.S. and Caldwell, D.G., *Origami-based robotic paper-and-board packaging for food industry.* Trends In Food Science & Technology, 2010. **21**(3): p. 153-157.
7. Kumar, V., Leonard, N.E., and Morse, A.S., *Cooperative control : 2003 Block Island workshop on cooperative control*. 2004, Berlin ; [London]: Springer. xii, 289 p.
8. Arai, T., Pagello, E., and Parker, L.E., *Advances in multirobot systems - Guest editorial.* Ieee Transactions on Robotics and Automation, 2002. **18**(5): p. 655-661.
9. IFR. *International Federation of Robotics*. 2014 [cited 21 Aug 2014]; Available from: http://www.ifr.org.
10. Craig, J.J. and Raibert, M.H. *A systematic method of hybrid position/force control of a manipulator*. in *Computer Software and Applications Conference, 1979. Proceedings. COMPSAC 79. The IEEE Computer Society's Third International*. 1979.
11. Ishida, T. *Force Control in Coordination of Two Arms*. in *Proc. 5th Int. Conf. on Artificial Intelligence*. 1977.
12. Siciliano, B. and Khatib, O., *Springer handbook of robotics*. 2008, Berlin: Springer. lx, 1611 p.
13. Lewis, F.L., Dawson, D.M., and Abdallah, C., *Robot manipulator control : theory and practice*. 2nd ed. Control engineering series ;. 2004, New York: Marcel Dekker. xvii, 614 p.
14. Merlet, J., *Parallel Manipulators - State-Of-The-Art and Perspectives.* Advanced Robotics, 1994. **8**(6): p. 589-596.
15. Uchiyama, M., *Structures And Characteristics Of Parallel Manipulators.* Advanced Robotics, 1994. **8**(6): p. 545-557.
16. Hui, C., Yiu-Kuen, Y., and Zexiang, L., *Dynamics and control of redundantly actuated parallel manipulators.* Mechatronics, IEEE/ASME Transactions on, 2003. **8**(4): p. 483-491.
17. Huang, H.-P. and Chen, R.-S., *Modeling and adaptive coordination control of a two-Robot system.* Journal of Robotic Systems, 1992. **9**(1): p. 65-92.
18. Desai, J.P., *Motion Planning and Control of Cooperative Robotic Systems*, Doctor of Philosophy. 1998, University of Pennsylvania.

19. Liu, J.-F. and Abdel-Malek, K., *Robust control of planar dual-arm cooperative manipulators.* Robotics and Computer-Integrated Manufacturing, 2000. **16**(2–3): p. 109-119.
20. Ghasemi, M. and Sadigh, M., *A direct algorithm to compute the switching curve for time-optimal motion of cooperative multi-manipulators moving on a specified path.* Advanced Robotics, 2008. **22**(5): p. 493-506.
21. Baxter, J.L., et al., *Shared Potential Fields and their place in a multi-robot co-ordination taxonomy.* Robotics and Autonomous Systems, 2009. **57**(10): p. 1048-1055.
22. Yu, J., Fang, Q., and Ke, Y., *Trajectory planning of multi-robot coordination platform for locating large subassembly.* International Journal of Modelling, Identification and Control, 2009. **6**(4): p. 357-366.
23. Seung Jun, L. and Seul, J. *Experimental studies of an object handling task by force control between two balancing robots*. in *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*. 2011.
24. Walker, I.D., Freeman, R.A., and Marcus, S.I., *Internal Object Loading for Multiple Cooperating Robot Manipulators.* Proceedings - 1989 Ieee International Conference on Robotics and Automation, Vol 1-3, 1989: p. 606-611.
25. Nokleby, S., et al., *Force capabilities of redundantly-actuated parallel manipulators.* Mechanism and machine theory, 2005. **40**(5): p. 578-599.
26. Adli, M.A. and Hanafusa, H., *Contribution of Internal Forces to the Dynamics of Closed Chain Mechanisms.* Robotica, 1995. **13**: p. 507-514.
27. Muller, A., *Internal Preload Control of Redundantly Actuated Parallel Manipulators - Its Application to Backlash Avoiding Control.* Robotics, IEEE Transactions on, 2005. **21**(4): p. 668-677.
28. Tzamtzi, M.P., Koumboulis, F.N., and Skarpetis, M.G., *On the control of the liquid pouring process using cooperative manipulators.* International Journal of Modelling, Identification and Control, 2011. **12**(Compendex): p. 173-181.
29. Uchiyama, M., Iwasawa, N., and Hakomori, K. *Hybrid position/Force control for coordination of a two-arm robot*. in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*. 1987.
30. Hayati, S. *Hybrid position/Force control of multi-arm cooperating robots*. in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*. 1986.
31. Augugliaro, F., et al., *The Flight Assembled Architecture installation: Cooperative construction with flying machines.* Control Systems, IEEE, 2014. **34**(4): p. 46-64.
32. Rojas, J. and Peters, R., *Analysis of autonomous cooperative assembly using coordination schemes by heterogeneous robots using a control basis approach.* Autonomous Robots, 2012. **32**(4): p. 369-383.
33. Smith, C., et al., *Dual arm manipulation—A survey.* Robotics and Autonomous Systems, 2012. **60**(10): p. 1340-1353.
34. Wang, J., Dodds, S.J., and Bailey, W.N., *Co-ordinated control of multiple robotic manipulators handling a common object-theory and experiments.* Control Theory and Applications, IEE Proceedings -, 1997. **144**(1): p. 73-86.
35. Montemayor, G. and Wen, J.T., *Decentralized collaborative load transport by multiple robots.* 2005 IEEE International Conference on Robotics and Automation (ICRA), Vols 1-4, 2005: p. 372-377.

36. Simmons, R., et al., *First Results in the Coordination of Heterogeneous Robots for Large-Scale Assembly*, in *Experimental Robotics VII*, D. Rus and S. Singh, Editors. 2001, Springer Berlin / Heidelberg. p. 323-332.
37. Cao, Y.U., Fukunaga, A.S., and Kahng, A., *Cooperative Mobile Robotics: Antecedents and Directions.* Autonomous Robots, 1997. **4**(1): p. 7-27.
38. Dudek, G., Jenkin, M., and Milios, E., *A Taxonomy of Multirobot Systems*, in *Robot Teams*, T. Balch, Parker L.E., A K Peters, Editor. 2002.
39. Gerkey, B.P. and Matarić, M.J., *A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems.* The International journal of robotics research, 2004. **23**(9): p. 939-954.
40. Winfield, A.F.T., *Towards an Engineering Science of Robot Foraging*. Distributed Autonomous Robotic Systems 8, ed. H. Asama, et al. 2009, Berlin: Springer-Verlag Berlin. 185-192.
41. Farinelli, A., Iocchi, L., and Nardi, D., *Multi-robot systems: A classification focused on coordination.* IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2004. **34**(5): p. 2015-2028.
42. De Schutter, J., et al., *Force control: A bird's eye view*. 1997. p. 1-17.
43. Zeng, G. and Hemami, A., *An overview of robot force control.* Robotica, 1997. **15**(05): p. 473-482.
44. Salisbury, J.K. *Active stiffness control of a manipulator in cartesian coordinates*. in *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*. 1980.
45. Hogan, N., *Impedance Control: An Approach to Manipulation: Part I - Theory.* Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, 1985. **107**(Compendex): p. 1-7.
46. Schneider, S.A. and Cannon, R.H., Jr., *Object impedance control for cooperative manipulation: theory and experimental results.* Robotics and Automation, IEEE Transactions on, 1992. **8**(3): p. 383-394.
47. Bonitz, R.C. and Hsia, T.C., *Internal force-based impedance control for cooperating manipulators.* Robotics and Automation, IEEE Transactions on, 1996. **12**(1): p. 78-89.
48. Tsuji, T. and Kaneko, M., *Noncontact impedance control for redundant manipulators.* IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans., 1999. **29**(Compendex): p. 184-193.
49. Wimbock, T., Ott, C., and Hirzinger, G., *Impedance behaviors for two-handed manipulation: Design and experiments.* Proceedings of the 2007 Ieee International Conference on Robotics and Automation, Vols 1-10, 2007: p. 4182-4189.
50. Caccavale, F., et al., *Six-DOF Impedance Control of Dual-Arm Cooperative Manipulators.* Mechatronics, IEEE/ASME Transactions on, 2008. **13**(5): p. 576-586.
51. Raibert, M.H. and Craig, J.J., *Hybrid Position/Force Control Of Manipulators.* Journal of dynamic systems, measurement, and control, 1981. **103**(2): p. 126-133.
52. Yoshikawa, T., *Dynamic hybrid position/force control of robot manipulators-Description of hand constraints and calculation of joint driving force.* Robotics and Automation, IEEE Journal of, 1987. **3**(5): p. 386-392.
53. Ohishi, K., Miyazaki, M., and Fujita, M. *Hybrid control of force and position without force sensor*. in *Industrial Electronics, Control, Instrumentation, and*

Automation, 1992. Power Electronics and Motion Control., Proceedings of the 1992 International Conference on. 1992.

54. Perdereau, V. and Drouin, M., *Hybrid external control for two robot coordinated motion.* Robotica, 1996. **14**: p. 141-153.
55. Di, X., et al., *Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment.* Control Systems Technology, IEEE Transactions on, 2000. **8**(4): p. 635-645.
56. Kim, B., Song, J., and Park, J., *A Serial-Type Dual Actuator Unit With Planetary Gear Train: Basic Design and Applications.* IEEE/ASME Transactions on Mechatronics, 2010. **15**(1): p. 108-116.
57. Komada, S., Ohnishi, K., and Hori, T. *Hybrid position/force control of robot manipulators based on acceleration controller*. in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation, April 9, 1991 - April 11, 1991*. 1991. Sacramento, CA, USA: Publ by IEEE.
58. Anderson, R.J. and Spong, M.W., *Hybrid Impedance Control of Robotic Manipulators.* Ieee Journal of Robotics and Automation, 1988. **4**(5): p. 549-556.
59. Volpe, R. and Khosla, P., *A Theoretical and Experimental Investigation of Explicit Force Control Strategies for Manipulators.* Ieee Transactions on Automatic Control, 1993. **38**(11): p. 1634-1650.
60. Wang, H. and Xie, Y., *Adaptive Jacobian Force/Position Tracking Control of Robotic Manipulators in Compliant Contact with an Uncertain Surface.* Advanced Robotics, 2009. **23**(1): p. 165-183.
61. Rodriguez-Angeles, A., De J. Portillo-Velez, R., and Cruz-Villar, C.A. *An optimal admittance reactive force control for cooperative robot grasping tasks*. in *8th International Conference on Informatics in Control, Automation and Robotics, ICINCO 2011, July 28, 2011 - July 31, 2011*. 2011. Noordwijkerhout, Netherlands: INSTICC Press.
62. Wen-Hong, Z. and Salcudean, S.E., *Stability guaranteed teleoperation: an adaptive motion/force control approach.* Automatic Control, IEEE Transactions on, 2000. **45**(11): p. 1951-1969.
63. de Queiroz, M.S., et al., *Adaptive position/force control of robot manipulators without velocity measurements: theory and experimentation.* IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 1997. **27**(Compendex): p. 796-809.
64. Jean, J.H. and Fu, L.C., *An Adaptive-Control Scheme for Coordinated Multimanipulator Systems.* Ieee Transactions on Robotics and Automation, 1993. **9**(2): p. 226-231.
65. Deghat, M., Khayatian, A.R., and Eghtesad, M., *Experimental Study of a Robust-Adaptive Controller Design for two Cooperating RLED Robot Manipulators Carrying a Rigid Payload.* 2009 Ieee International Conference on Mechatronics, Vols 1 and 2, 2009: p. 83-88.
66. Oda, N., et al. *A robust impedance control strategy for redundant manipulator*. in *Industrial Electronics, Control, and Instrumentation, 1995., Proceedings of the 1995 IEEE IECON 21st International Conference on*. 1995.
67. Oh, Y., Chung, W.K., and Suh, I.H. *Disturbance observer based robust impedance control of redundant manipulators*. in *1999 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99): Human and Environment Friendly Robots whith High Intelligence and Emotional Quotients', October 17, 1999 - October 21, 1999*. 1999. Kyongju, South Korea: IEEE.

68. Albu-Schaffer, A., Ott, C., and Hirzinger, G., *A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots.* The International journal of robotics research, 2007. **26**(1): p. 23-39.
69. Mohajerpoor, R., et al., *A robust adaptive hybrid force/position control scheme of two planar manipulators handling an unknown object interacting with an environment.* Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2012. **226**(4): p. 509-522.
70. Kawamura, S., Miyazaki, F., and Arimoto, S. *Hybrid Position/Force Control Of Robot Manipulators Based On Learning Method*. in *Proceedings of '85 International Conference on Advanced Robotics.* 1985. Tokyo, Jpn: Japan Industrial Robot Assoc.
71. Cheah, C.-C. and Wang, D., *Learning impedance control for robotic manipulators.* IEEE transactions on robotics and automation, 1998. **14**(Compendex): p. 452-465.
72. Pandian, S.R. and Kawamura, S., *Hybrid force position control for robot manipulators based on a D-type learning law.* Robotica, 1996. **14**: p. 51-59.
73. Buchli, J., et al., *Learning variable impedance control.* The International Journal of Robotics Research, 2011. **30**(7): p. 820-833.
74. Kwan, C.M., Yesildirek, A., and Lewis, F.L. *Robust force/motion control of constrained robots using neural network*. in *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*. 1994.
75. Lin, F.J., Chou, P.H., and Kung, Y.S., *Robust fuzzy neural network controller with nonlinear disturbance observer for two-axis motion control system.* IET Control Theory & Applications, 2008. **2**(2): p. 151-167.
76. Baptista, L.F., Sousa, J.M., and Da Costa, J.S., *Force control of robotic manipulators using a fuzzy predictive approach.* Journal of Intelligent & Robotic Systems, 2001. **30**(4): p. 359-376.
77. Gueaieb, W., Karray, F., and Al-Sharhan, S., *A Robust Hybrid Intelligent Position/Force Control Scheme for Cooperative Manipulators.* Mechatronics, IEEE/ASME Transactions on, 2007. **12**(2): p. 109-125.
78. Khatib, O., *A unified approach for motion and force control of robot manipulators: The operational space formulation.* Robotics and Automation, IEEE Journal of, 1987. **3**(1): p. 43-53.
79. Hogan, N., *Impedance Control: An Approach to Manipulation: Part II - Implementation.* Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, 1985. **107**(Compendex): p. 8-16.
80. Hogan, N., *Impedance Control: An Approach to Manipulation: Part III - Applications.* Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, 1985. **107**(Compendex): p. 17-24.
81. Carignan, C.R. and Akin, D.L., *Cooperative control of two arms in the transport of an inertial load in zero gravity.* Robotics and Automation, IEEE Journal of, 1988. **4**(4): p. 414-419.
82. Leksono, E., Murakami, T., and Ohnishi, K. *Cooperative motion control of multimanipulator based on workspace disturbance observer with variable compliance gain*. in *Industrial Electronics, Control and Instrumentation, 1997. IECON 97. 23rd International Conference on*. 1997.
83. Tinos, R., Terra, M.H., and Ishihara, J.Y., *Motion and force control of cooperative robotic manipulators with passive joints.* Ieee Transactions on Control Systems Technology, 2006. **14**(4): p. 725-734.

84. Jankowski, K.P. and Elmaraghy, H.A., *Dynamic Decoupling for Hybrid Control of Rigid-Joint Flexible-Joint Robots Interacting with the Environment.* Ieee Transactions on Robotics and Automation, 1992. **8**(5): p. 519-534.
85. Lange, F., et al. *Revised force control using a compliant sensor with a position controlled robot*. in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. 2012.
86. Dai, J.S. and Kerr, D.R., *A six-component contact force measurement device based on the Stewart platform.* Institution of Mechanical Engineers. Proceedings. Part C: Journal of Mechanical Engineering Science, 2000. **214**(5): p. 687-697.
87. Lopes, A. and Almeida, F., *A force-impedance controlled industrial robot using an active robotic auxiliary device.* Robotics and Computer-Integrated Manufacturing, 2008. **24**(3): p. 299-309.
88. Khatib, O. and Burdick, J. *Motion and Force Control of Robot Manipulators*. in *Proceedings - 1986 IEEE International Conference on Robotics and Automation.* 1986. San Francisco, CA, USA: IEEE.
89. Khatib, O., *The Operational Space Framework.* JSME international journal. Ser. C, Dynamics, control, robotics, design and manufacturing, 1993. **36**(3): p. 277-287.
90. Yoshikawa, T., Sugie, T., and Tanaka, M., *Dynamic hybrid position/force control of robot manipulators-controller design and experiment.* Robotics and Automation, IEEE Journal of, 1988. **4**(6): p. 699-705.
91. Yoshikawa, T. and Zheng, X.-Z., *Coordinated Dynamic Hybrid Position/Force Control for Multiple Robot Manipulators Handling One Constrained Object.* The International journal of robotics research, 1993. **12**(3): p. 219-230.
92. Wu, J., Wang, J., and You, Z., *An overview of dynamic parameter identification of robots.* Robotics and Computer-Integrated Manufacturing, 2010. **26**(5): p. 414-419.
93. Chae, H.A., Christopher, G.A., and John, M.H. *Estimation of inertial parameters of rigid body links of manipulators*. in *Decision and Control, 1985 24th IEEE Conference on*. 1985.
94. Reyes, F. and Chavez, C., *Experimental evaluation of parameter identification schemes on a direct-drive robot.* Journal of Systems and Control Engineering, 2012. **226**(10): p. 1419-1431.
95. Pintelon, R. and Schoukens, J., *System identification a frequency domain approach*. 2012, Wiley; IEEE Press: Hoboken, N.J.; Piscataway, NJ. p. xliv, 743 p.
96. Gautier, M., Briot, S., and Venture, G., *Identification of Consistent Standard Dynamic Parameters of Industrial Robots.* 2013 Ieee/Asme International Conference on Advanced Intelligent Mechatronics (Aim): Mechatronics for Human Wellbeing, 2013: p. 1429-1435.
97. Khalil, W. and Dombre, E., *Modeling, identification & control of robots*. Kogan Page Science paper ed. 2004, London: Kogan Page Science. xix, 480 p.
98. Armstrong, B., *On Finding Exciting Trajectories for Identification Experiments Involving Systems with Nonlinear Dynamics.* The International Journal of Robotics Research, 1989. **8**(6): p. 28-48.
99. Gautier, M. and Khalil, W., *Exciting Trajectories for the Identification of Base Inertial Parameters of Robots.* The International Journal of Robotics Research, 1992. **11**(4): p. 362-375.
100. Swevers, J., Verdonck, W., and De Schutter, J., *Dynamic Model Identification for Industrial Robots.* Control Systems Magazine, IEEE, 2007. **27**(5): p. 58-71.

101. Swevers, J., et al., *Optimal robot excitation and identification.* Robotics and Automation, IEEE Transactions on, 1997. **13**(5): p. 730-740.
102. Spong, M.W., Hutchinson, S., and Vidyasagar, M., *Robot modeling and control*. 2006, Hoboken, N.J.: John Wiley & Sons. 478 p.
103. Sahinkaya, M.N., *On inverse dynamics simulation.* 2012.*Personal Communication*
104. Sahinkaya, M.N., *Inverse dynamic analysis of multiphysics systems.* Proceedings of the Institution of Mechanical Engineers Part I-Journal of Systems and Control Engineering, 2004. **218**(I1): p. 13-26.
105. Sahinkaya, M.N., *Input shaping for vibration-free positioning of flexible systems.* Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering, 2001. **215**(Compendex): p. 467-481.
106. Sahinkaya, M.N., *Virtual non-linear disturbance observer by dual inverse dynamic modelling.* Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2007. **221**(Compendex): p. 677-688.
107. Du, Z., Iravani, P., and Sahinkaya, M.N. *A New Approach to Design Optimal Excitation Trajectories for Parameter Estimation of Robot Dynamics*. in *Control, UKACC 10th International Conference on*. 2014. Loughborough, UK.
108. Geomagic. *The Geomagic® Touch™ Haptic Device (formerly Sensable Phantom Omni)*. 2014 [cited 31 Mar 2014]; Available from: http://geomagic.com/en/products/phantom-omni/overview.
109. Quanser. *QUARC*. 2010 [cited 31 Mar 2014]; Available from: http://www.quanser.com/Products/quarc.
110. Shimada, N., et al. *Smooth touch and force control to unknown environment without force sensor for industrial robot*. in *Advanced Motion Control, 2010 11th IEEE International Workshop on*. 2010.
111. LTD, A.M. *Miniature S-Beam Load Cell | DBBSMM*. 2013 [cited 01 Oct 2014]; Available from: http://www.appmeas.co.uk/dbbsmm-miniature-s-beam-load-cell.html.
112. Gautier, M. and Khalil, W., *Direct Calculation of Minimum Set of Inertial Parameters of Serial Robots.* Ieee Transactions on Robotics and Automation, 1990. **6**(3): p. 368-373.
113. Schroeder, M., *Synthesis of low-peak-factor signals and binary sequences with low autocorrelation (Corresp.).* Information Theory, IEEE Transactions on, 1970. **16**(1): p. 85-89.
114. Pintelon, R., Schoukens, J., and John Wiley & Sons., *System identification a frequency domain approach*. 2001, IEEE Press: New York. p. xxxviii, 605 p.
115. Naerum, E., Cornella, J., and Elle, O.J. *Contact force estimation for backdrivable robotic manipulators with coupled friction*. in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. 2008.
116. Spong, M.W., Hutchinson, S., and Vidyasagar, M., *Robot dynamics and control*. Second Edition ed. 2004: Wiley. 303p.
117. Haugen, F., *Comparing PI Tuning Methods in a Real Benchmark Temperature Control System.* Modeling Identification and Control, 2010. **31**(3): p. 79-91.
118. Mathwork, *Matlab*. 2009.
119. Du, Z., Sahinkaya, M.N., and Iravani, P. *Combined Force and Position Controller Based on Inverse Dynamics: Application to Cooperative Robotics*. in *Towards Autonomous Robotic Systems 15th Annual Conference TAROS*. 2014. Birmingham, UK: Springer.

Appendix A

Dysim Program Demonstration

To demonstrate how Dysim program works, the constrained two-link manipulator in Section 3.1.3 is used as an example. First of all, there are two kinds of user interfaces: the Default Interface and the Planer (2D) Mechanism Interface. In the Default Interface, the user enters the Lagrangian function, constraint Jacobians, and constraint equations. In the Planer (2D) Mechanism Interface, the user specifies parameters for each element and connection of a two-dimensional mechanism, and then the interface automatically derives the Lagrangian function, constraint Jacobians, and constraint equations. The Planer Mechanism Interface is demonstrated here.

First, the mass and inertia of the two links and the mass of the load are created. Then their connections are specified in terms of each object's local coordinates (Figure A.1). The system variables, initial conditions, the Lagrangian function, constraint Jacobians and constraint equations are then derived automatically (Figure A.2 to Figure A.5). Next, in the case of inverse dynamics, motion input coordinates need to be specified (Figure A.6). Force input can also be specified on according constraints. Finally, command files can be generated for the use in Simulink (Figure A.0.7).

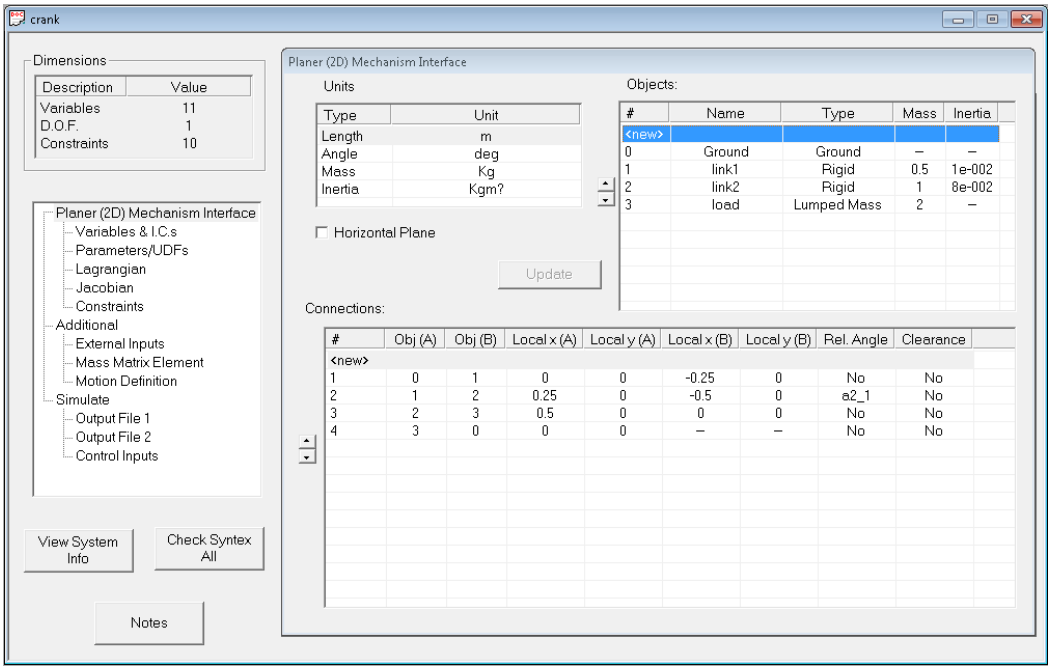

Figure A.1 Dysim program interface: objects and connections

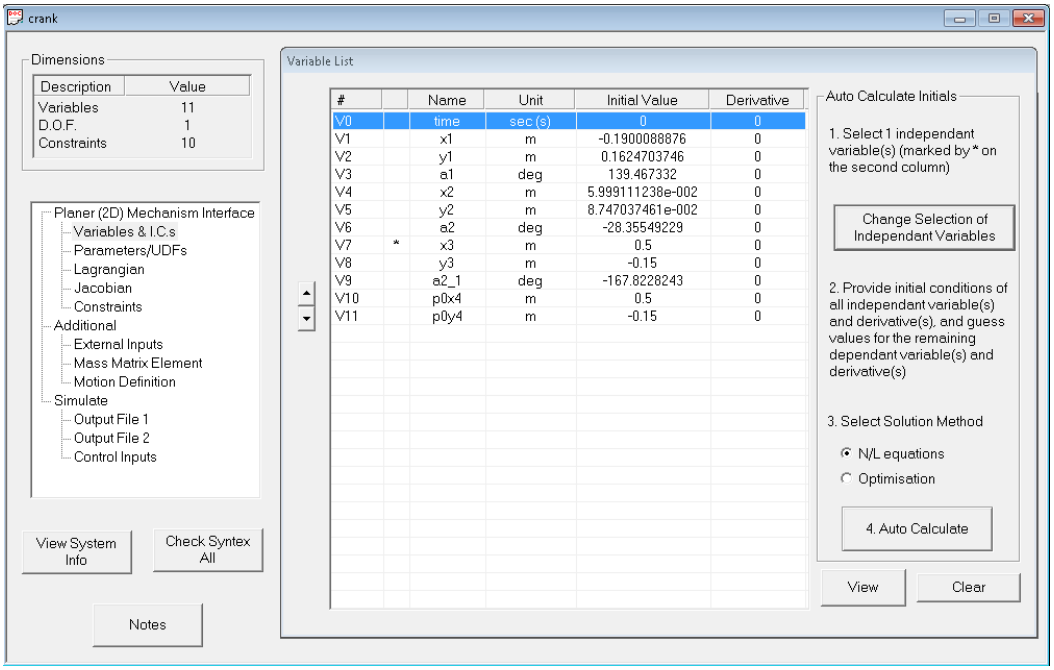

Figure A.2 Dysim program interface: variables and initial conditions

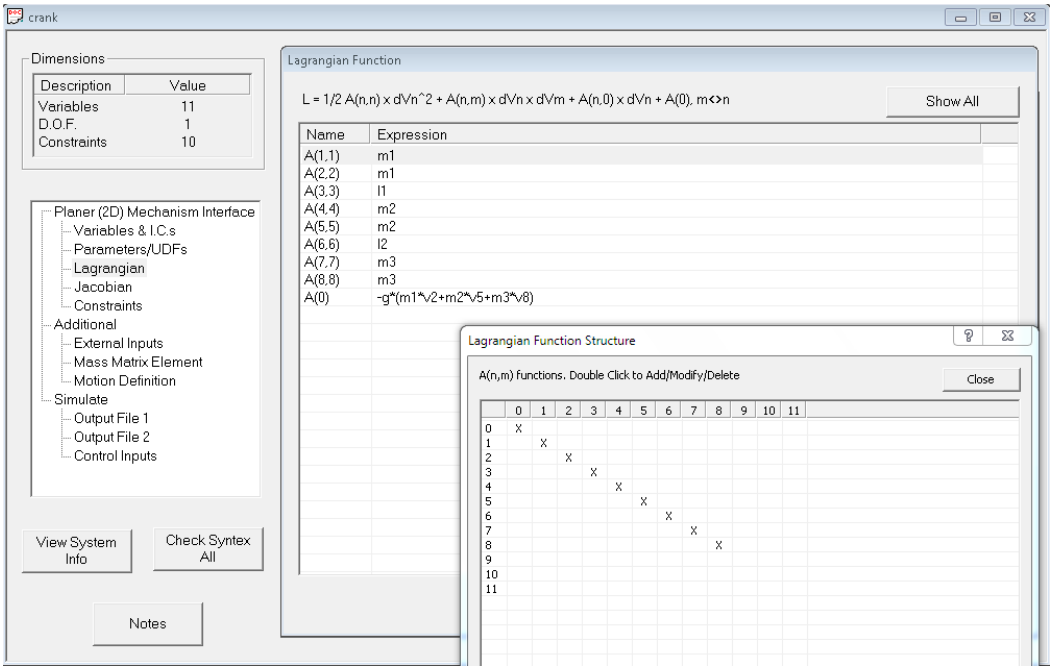

Figure A.3 Dysim program interface: Lagrangian functions

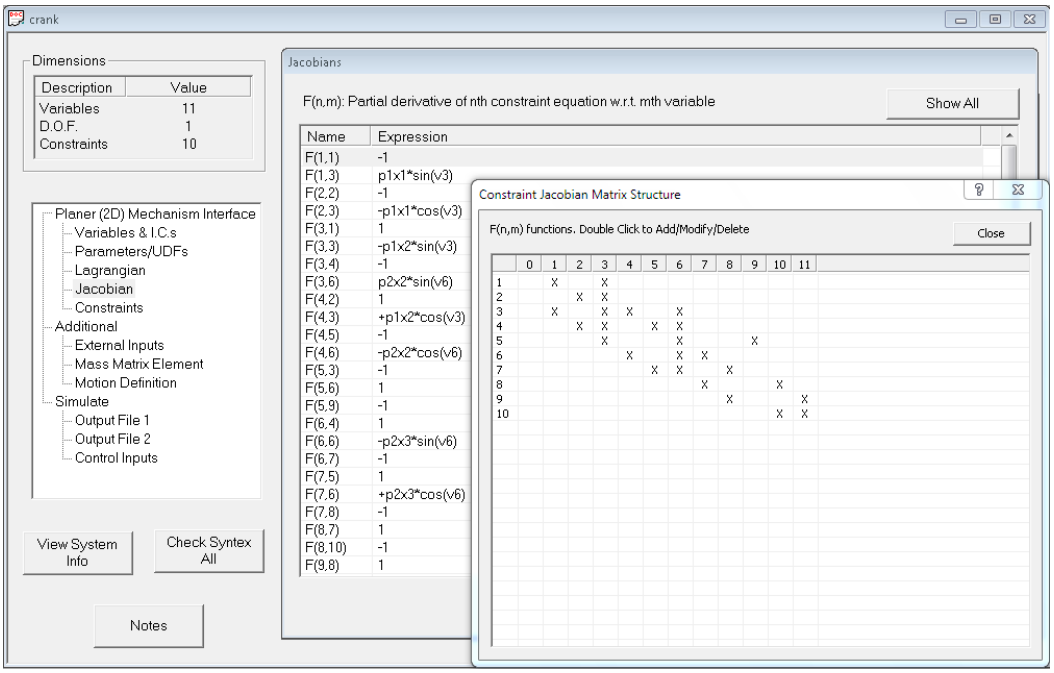

Figure A.4 Dysim program interface: constraint Jacobians

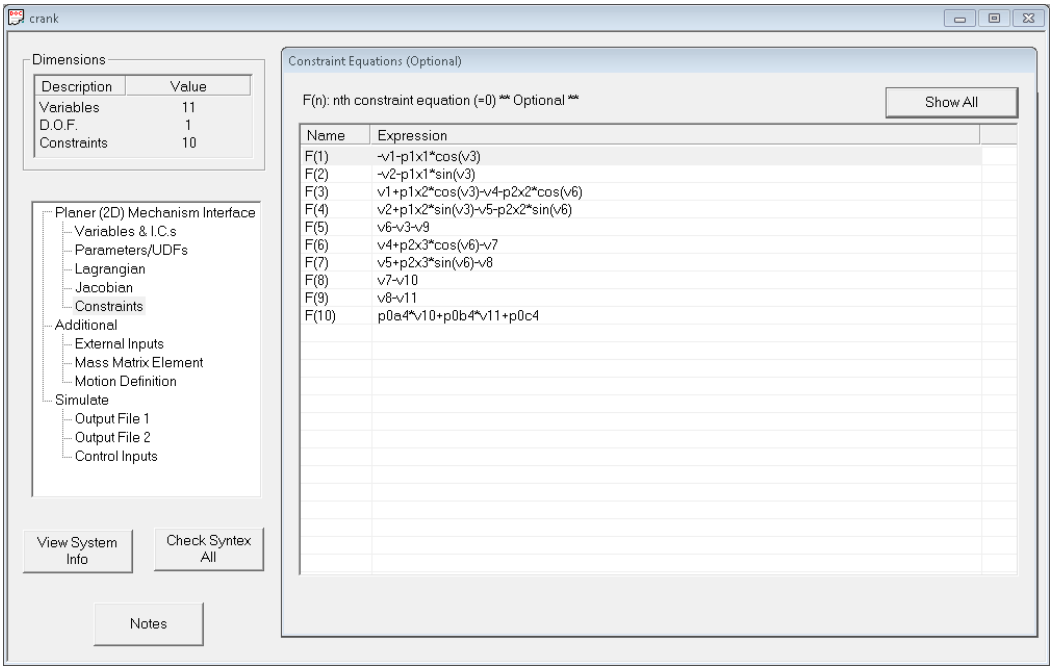

Figure A.5 Dysim program interface: constraint functions

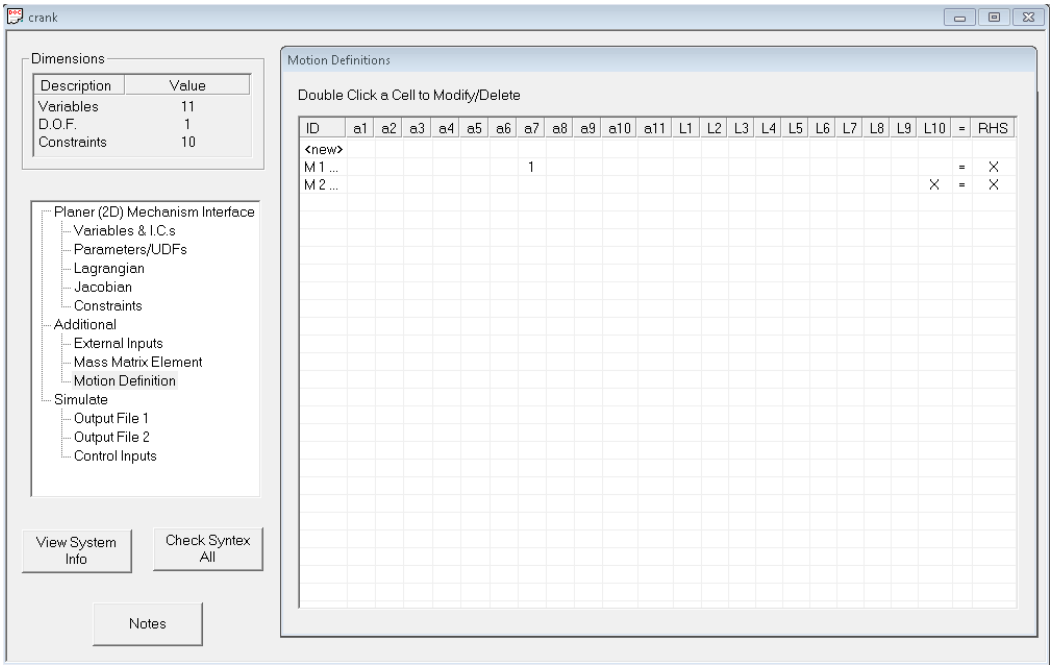

Figure A.6 Dysim program interface: motion definition (including constraint force)

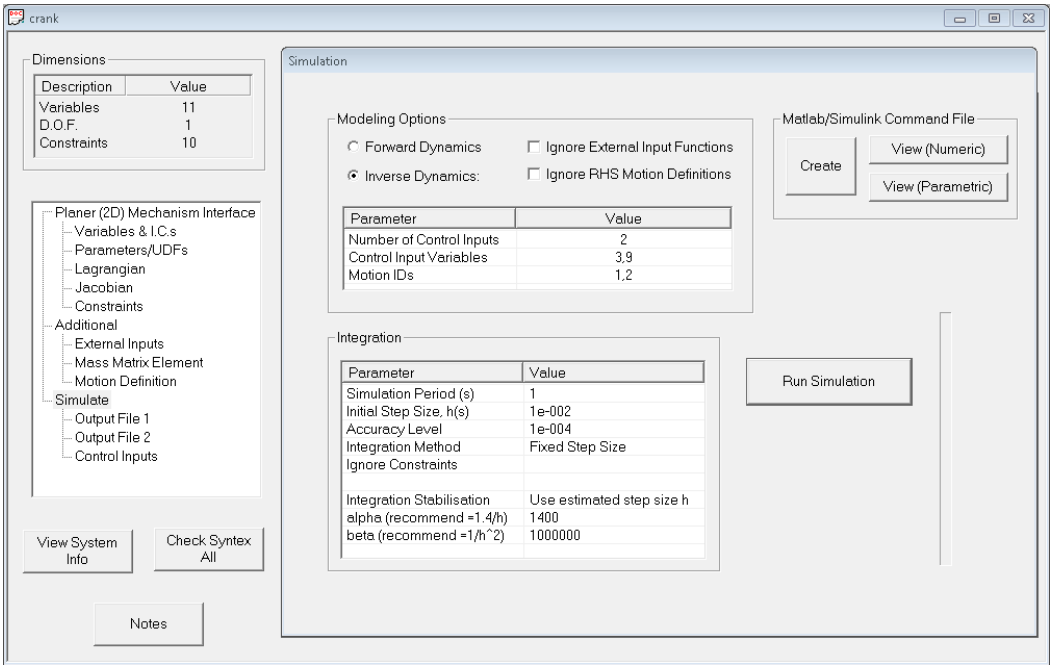

Figure A.0.7 Dysim program interface: model output

Figure A.8 shows the Dysim block in the Simulink environment. User can choose the input and output coordinates, and the constraint force (Figure A.9).

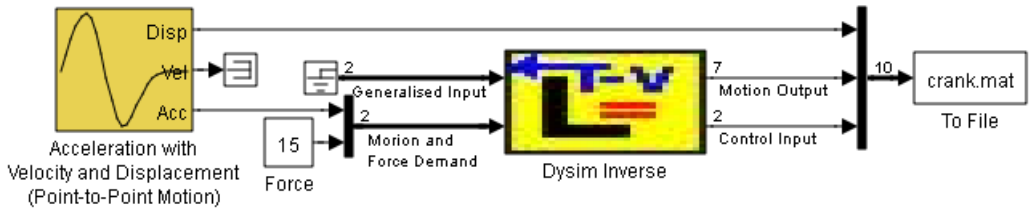

Figure A.8 Dysim Inverse Simulink block

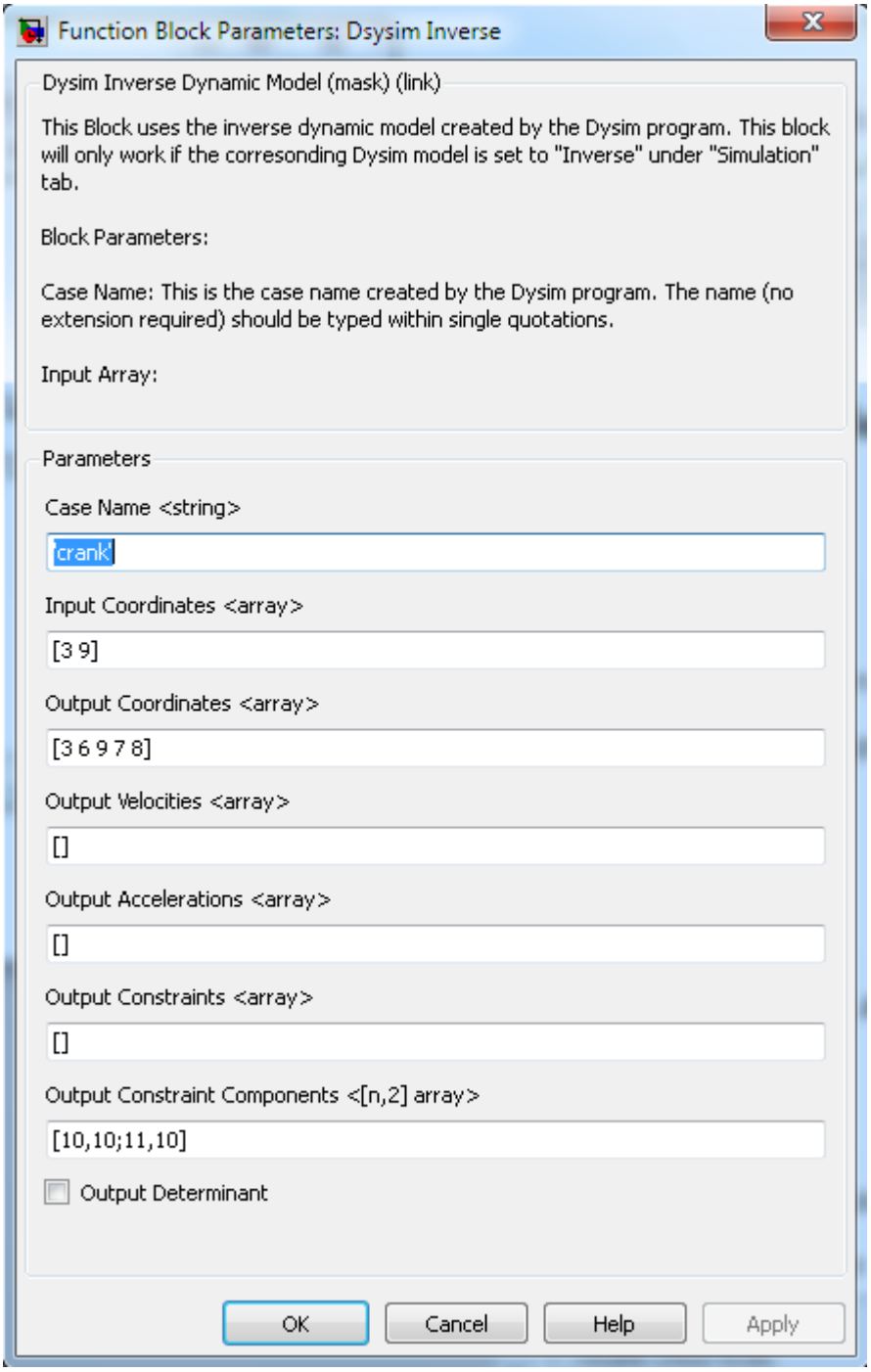

Figure A.9 Dysim Simulink block parameters